ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

Πολυτεχνική Σχολή

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ Η/Υ

Ανάπτυξη εφαρμογής Android “Προσωπική Ψηφιακή Ατζέντα”

Android application development: “Personal Digital Agenda”

Διπλωματική Εργασία

ΜΑΓΓΑΣΙΑΝΟΣ ΝΙΚΟΛΑΟΣ

Επιβλέποντες Καθηγητές: Αλκιβιάδης Ακρίτας, Καθηγητής

Γεώργιος Σταμούλης, Καθηγητής

Βόλος, 2016

Περίληψη

Η εξαιρετικά ραγδαία ανάπτυξη την τεχνολογίας και κατ' επέκταση των κινητών τηλεφώνων έχουν εισέλθει για τα καλά στην καθημερινή ζωή των ανθρώπων. Πλέον, πολλές ανάγκες μας ικανοποιούνται σε πολύ μικρό χρονικό διάστημα και χωρίς ιδιαίτερο κόπο σε σχέση με τα παλαιότερα χρόνια. Μπορούμε να ακούμε μουσική, να βγάζουμε φωτογραφίες, να διαβάζουμε εφημερίδες, να ελέγχουμε τα emails μας, και να κάνουμε άλλες καθημερινές μας συνήθειες με την χρήση μόνο του κινητού μας τηλεφώνου.

Στην παρούσα διπλωματική εργασία θα γίνει αναλυτική περιγραφή και οπτική παρουσίαση μιας εφαρμογής ψηφιακής ατζέντας που θα απευθύνεται σε χρήστες στην κατοχή των οποίων υπάρχουν συσκευές Android.

Abstract

The rapid development of technology and by extension the mobile phones have entered for good in peoples' lives. Many of our needs met in a very short time and without effort compared to the earlier years. We can listen to music, take pictures, read newspapers, to check our emails, and do other daily habits using only our mobile phone.

This thesis will be a detailed description and visual presentation of a digital agenda application targeted at users in possession of which are Android devices.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον καθηγητή κ. Αλκιβιάδη Ακρίτα που μου έδωσε την δυνατότητα να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα. Τον ευχαριστώ για τη βοήθεια, καθοδήγηση και τις χρήσιμες συμβουλές που μου παρείχε καθ' όλη την διάρκεια εκπόνησής της.

Θα ήθελα επίσης να απευθύνω τις ευχαριστίες μου στους γονείς μου, οι οποίοι στήριζαν τις σπουδές μου με διάφορους τρόπους, φροντίζοντας για την καλύτερη δυνατή μόρφωση μου.

Τέλος, ένα μεγάλο ευχαριστώ στους φίλους μου, για την στήριξη και τις συμβουλές τους όλα αυτά τα χρόνια των σπουδών μου.

Περιεχόμενα

Κεφάλαιο 1

1 Εισαγωγή

1.1 Περιγραφή της εφαρμογής

Η εργασία αυτή περιγράφει όλα τα βήματα για την ανάπτυξη και τον σχεδιασμό εφαρμογής για έξυπνα τηλέφωνα (smartphones) σε λειτουργικό σύστημα Android. Το όνομα της εφαρμογής είναι Personal Digital Agenda και απευθύνεται στους ανθρώπους που δυσκολεύονται να θυμούνται διάφορα γεγονότα και εορτές, δίνοντας τους την δυνατότητα να τα σημειώνουν στην εφαρμογή και αυτή να τους τα θυμίζει όποτε αυτοί το έχουν ορίσει. Επίσης, μπορούν να καταγράψουν γεγονότα και πράγματα που δεν είναι τόσο σημαντικά, όπως κάποιες εξωτερικές δουλειές ή κάποια λίστα για super market, χωρίς να τους έρθει κάποια ειδοποίηση από την εφαρμογή.

1.2 Γενικά για το Android

Το Android είναι ένα λειτουργικό σύστημα που ενσωματώνεται σε συσκευές κινητής τηλεφωνίας, τα οποία διαθέτουν οθόνη αφής, τρέχουν τον πυρήνα kernel του λειτουργικού Linux και ακόμη, επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με τη χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας τη συσκευή μέσω βιβλιοθηκών λογισμικού της Google.

Συσκευές με Android υπάρχουν πλέον πάρα πολλές, η καθεμία με διαφορετικά χαρακτηριστικά και από διάφορες κατασκευάστριες εταιρίες. Το πολύ θετικό με τις συσκευές Android είναι ότι είναι αφενός multimedia συσκευές, δηλαδή σας δίνουν τη δυνατότητα να αναπαράγετε πολλαπλά μέσα και multitasking, δηλαδή σας δίνουν τη δυνατότητα εκτέλεσης πολλών εφαρμογών ταυτόχρονα, π.χ. ακούτε τραγούδια ενώ σερφάρετε στο ίντερνετ και ταυτόχρονα απαντάτε σε

ένα SMS χωρίς να κλείσετε καμία εφαρμογή ή να χάσετε τη σελίδα που επισκεφτήκατε.

Βασικό χαρακτηριστικό του Android, επίσης, είναι η πληθώρα εφαρμογών που υπάρχουν στο Play Store δωρεάν και μη, όπου μπορεί ο χρήστης να κατεβάσει.

Όσον αφορά το hardware οι διπύρηνοι, τετραπύρηνοι ή και οκταπύρηνοι πλέον επεξεργαστές και οι διακεκριμένες GPU είναι πλέον γεγονός κάνοντας την εμπειρία χρήσης όλο και καλύτερη.

1.3 Εργαλεία που χρησιμοποιήθηκαν

Η εφαρμογή αναπτύχθηκε αποκλειστικά στο περιβάλλον που προσφέρει το Android Studio, το επίσημο IDE που παρέχεται για την ανάπτυξη εφαρμογών Android, και απευθύνεται σε συσκευές με έκδοση Android API μεγαλύτερη ή ίση του 11 οι οποίες και καλύπτουν πλέον το 96% των συσκευών παγκοσμίως. Ο λόγος για τον οποίο χρησιμοποιήθηκε το Android Studio είναι ότι αυτό το IDE έχει σχεδιαστεί ειδικά για ανάπτυξη σε Android, καθιστώντας το έτσι πιο γρήγορο και απλούστερο στην ανάπτυξη εφαρμογών. Επίσης, το Android Studio παρέχει κάποιες δυνατότητες όπως το Gradle-based σύστημα κατασκευής, η ενσωματωμένες λειτουργίες ελέγχου χρήσης την μνήμης και τις CPU, layout editor με υποστήριξη για drag and drop επεξεργασία, η εύκολη ενσωμάτωση emulators για την ασφαλέστερη δοκιμή της εφαρμογής, το built-in σύστημα Building και Compiling του κώδικα.

1.4 Δομή εργασίας

Η εργασία θα ξεκινήσει με τον ορισμό κάποιον εννοιών και οντοτήτων που χρησιμοποιήθηκαν στην εφαρμογή, έτσι ώστε να γίνει πιο εύκολη η κατανόηση της. Θα ακολουθήσει η ανάλυση των κλάσεων που έχουν οριστεί στην εφαρμογή. Η ανάλυση τους χωρίζεται σε κεφάλαια ανάλογα με την λειτουργία που υλοποιούν, ένα κεφάλαιο ανά λειτουργία. Παράλληλα με την περιγραφή των λειτουργιών θα

υπάρχουν και κατάλληλα στιγμιότυπα των λειτουργιών της εφαρμογής ώστε ο αναγνώστης να έχει την δυνατότητα οπτικής εξέτασης των όσων περιγράφονται.

Κεφάλαιο 2

2 Γενικές έννοιες

Πριν από την περιγραφή της υλοποίησης της εφαρμογής μας θα ήταν σημαντικό να ορίσουμε κάποιες έννοιες και οντότητες που θα χρησιμοποιηθούν παρακάτω.

Activity: Η κλάση Activity παρέχει μία οθόνη αλληλεπίδρασης με τον χρήστη μέσω της μεθόδου setContentView(View). Αυτή η μέθοδος καλείται μέσα στην συνάρτηση onCreate(Bundle), η οποία αρχικοποιεί το activity μας.

Fragment: Είναι ένα κομμάτι από την διεπαφή χρήστη μιας εφαρμογής, το οποίο τοποθετείται μέσα σε ένα activity και εξαρτάται άμεσα από αυτό. Η αλληλεπίδραση με τα fragments μέσα σε ένα activity γίνεται μέσα του FragmentManager. Στην εφαρμογή αυτή όλες οι κύριες επιλογές του χρήστη έχουν υλοποιηθεί ως διαφορετικά fragments, ο έλεγχος των διαφορετικών λειτουργιών γίνεται αλλάζοντας fragments αντί για να αλλάζουμε διαφορετικά activities.

Intent: Είναι μια αφηρημένη περιγραφή μιας λειτουργίας που πρέπει να εκτελεστεί. Το intent παρέχει μια υπηρεσία διεξαγωγής late binding μεταξύ κώδικα σε διαφορετικές εφαρμογές. Σημαντικότερη χρήση του είναι κατά την έναρξη των activities. Πρόκειται ουσιαστικά για μια παθητική δομή δεδομένων που κατέχει μια αφηρημένη περιγραφή της δράσης που πρέπει να εκτελεστεί.

Navigation Drawer: Είναι ένα πάνελ που εμφανίζει τις κύριες επιλογές πλοήγησης της εφαρμογής και εμφανίζεται στο αριστερό άκρο της οθόνης. Τις περισσότερες φορές είναι κρυμμένο αλλά εμφανίζεται όταν ο χρήστης σύρει το δάχτυλο του από την αριστερή άκρη και προς τα δεξιά ή αν πατήσει το εικονίδιο στο πάνω μέρος την εφαρμογής.

Adapter: Ένα αντικείμενο που χαρακτηρίζεται σαν adapter λειτουργεί σαν γέφυρα μεταξύ ενός AdapterView και των υποκείμενων δεδομένων για αυτό το συγκεκριμένο view. Ο Adapter είναι επίσης υπεύθυνος για την κατασκευή ενός View για κάθε στοιχείο στο σύνολο δεδομένων. Στην εφαρμογή μας χρησιμοποιούμε ArrayAdapter, ListAdapter, καθώς και κάποιος custom Adapters που ορίζουμε εμείς.

ArrayAdapter: Η κατηγορία ArrayAdapter μπορεί να χειριστεί μια λίστα ή πίνακα αντικειμένων Java ως είσοδο. Κάθε αντικείμενο αντιστοιχίζεται σε μια γραμμή και από προεπιλογή χαρτογραφεί την μέθοδο toString() του αντικειμένου σε μία προβολή στο layout των γραμμών.

ListAdapter: Περιέχει τον τρόπο που φαίνονται οι λίστες στον χρήστη.

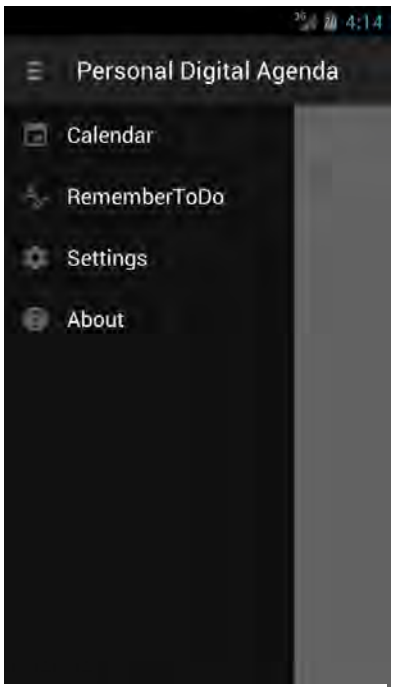
Option Menu: Είναι μια συλλογή από επιλογές για ένα activity, το οποίο εμφανίζεται όταν ο χρήστης αγγίξει (πατήσει) το κουμπί menu.

Κεφάλαιο 3

3 MainActivity και Navigation Drawer

3.1 MainActivity

Η κλάση MainActivity.java αποτελεί την κύρια κλάση της εφαρμογής. Σε αυτή την κλάση περιέχεται ο κώδικας όπου ορίζεται η βασική δομή και ο βασικός έλεγχος όλων των λειτουργιών της εφαρμογής. Η μέθοδος onCreate() χρησιμοποιείται για την αρχικοποίηση του activity, τον ορισμό των fragments που διαχειρίζονται την συμπεριφορά, τις αλληλεπιδράσεις και την παρουσίαση του navigation drawer. Μέσα στη MainActivity.java υπάρχουν ακόμα και κάποιες μέθοδοι για την αρχικοποίηση και συμπεριφορά του OptionMenu.



Εικόνα 3-1

3.1.1 onCreate() Method

Όπως φαίνεται και στον κώδικα που ακολουθεί παρακάτω, σε αυτή τη μέθοδο γίνεται η σύνδεση των μεταβλητών του navigation drawer με τα αντίστοιχα .xml αρχεία. Ακολουθεί ο ορισμός των ονομάτων των στοιχείων του συρταριού καθώς και ο ορισμός των εικονιδίων τους. Στη συνέχεια καλείται ο adapter που θα περιγραφτεί στην επόμενη παράγραφο για να προσαρμόσει αυτά τα στοιχεία (όνομα, εικονίδιο) στις γραμμές των στοιχείων του συρταριού.

Παρακάτω παρατίθεται ο κώδικας της onCreate().

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    mTitle = mDrawerTitle = getTitle();

    mNavigationDrawerItemTitles=
getResources().getStringArray(R.array.navigation_drawer_items_array);
    mDrawerLayout = (DrawerLayout) findViewById(R.id.drawer_layout);
    mDrawerList = (ListView) findViewById(R.id.left_drawer);


    ObjectDrawerItem[] drawerItem = new ObjectDrawerItem[4];

    drawerItem[0] = new ObjectDrawerItem(R.drawable.ic_calendar, "Calendar");
    drawerItem[1] = new ObjectDrawerItem(R.drawable.ic_remember2,
"RememberToDo");
    drawerItem[2] = new ObjectDrawerItem(R.drawable.ic_settings, "Settings");
    drawerItem[3] = new ObjectDrawerItem(R.drawable.ic_about, "About");

    DrawerItemCustomAdapter adapter = new DrawerItemCustomAdapter(this,
R.layout.listview_item_row, drawerItem);
    mDrawerList.setAdapter(adapter);

    mDrawerList.setOnItemClickListener(new DrawerItemClickListener());


    mDrawerLayout = (DrawerLayout) findViewById(R.id.drawer_layout);
    mDrawerToggle = new ActionBarDrawerToggle(this, mDrawerLayout,
R.drawable.ic_drawer, R.string.drawer_open,
            R.string.drawer_close) {

        /** Called when a drawer has settled in a completely closed state. */
        public void onDrawerClosed(View view) {
            super.onDrawerClosed(view);
            getSupportActionBar().setTitle(mTitle);
        }

        /** Called when a drawer has settled in a completely open state. */
        public void onDrawerOpened(View drawerView) {
            super.onDrawerOpened(drawerView);
            getSupportActionBar().setTitle(mDrawerTitle);
        }
    };

    mDrawerLayout.setDrawerListener(mDrawerToggle);
    mDrawerLayout.openDrawer(Gravity.LEFT);

    getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    getSupportActionBar().setHomeButtonEnabled(true);
    getSupportActionBar().setHomeAsUpIndicator(R.drawable.ic_drawer);
}
```

3.1.2 DrawerItemClickListener() Method

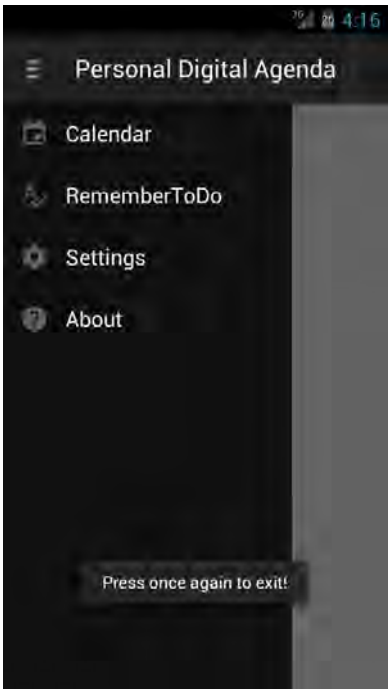
Αυτή η μέθοδος καλείται όταν ο χρήστης αγγίξει ένα αντικείμενο του navigation drawer. Μέσα σε αυτή καλείται η μέθοδος με όνομα selectItem(). Εκεί ανάλογα με το ποιο στοιχείο έχει επιλέξει ο χρηστής δημιουργείται και το ανάλογο fragment.

Ακολουθεί ο κώδικας της DrawerItemClickListener().

```
private class DrawerItemClickListener implements ListView.OnItemClickListener {

    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position,
long id) {
        selectItem(position);
    }
}

private void selectItem(int position) {

    Fragment fragment = null;


    switch (position) {
        case 0:
            fragment = new RadCalendarFragment();
            break;
        case 1:
            fragment = new RememberFragment();
            break;
        case 2:
            fragment = new SettingsFragment();
            break;
        case 3:
            fragment = new AboutFragment();
            break;

        default:
            break;
    }

    if (fragment != null) {
        FragmentManager fragmentManager = getSupportFragmentManager();
        fragmentManager.beginTransaction().replace(R.id.content_frame,
fragment).addToBackStack(null).commit();

        mDrawerList.setItemChecked(position, true);
        mDrawerList.setSelection(position);
        setTitle(mNavigationDrawerItemTitles[position]);
        mDrawerLayout.closeDrawer(mDrawerList);

    } else {
        Log.e("MainActivity", "Error in creating fragment");
    }
}
```

3.1.3 onBackPressed() Method

Εικόνα 3-2



Αυτή η μέθοδος καλείται όταν ο χρήστης πατήσει το κουμπί “πίσω”. Όταν συμβεί αυτό, εμφανίζεται ένα μήνυμα που λέει ότι πρέπει να το ξαναπατήσει για έξοδο («Press once again to exit!»). Αν το πατήσει για δεύτερη φορά μέσα σε 2 δευτερόλεπτα τότε βγαίνει από την εφαρμογή.

Ο κώδικας της μεθόδου είναι ως εξής.

```
@Override
public void onBackPressed()
{
    if (back_pressed + 2000 > System.currentTimeMillis())
        super.onBackPressed();
    else
        Toast.makeText(getBaseContext(), "Press once again to exit!",
Toast.LENGTH_SHORT).show();

    back_pressed = System.currentTimeMillis();
}
```

3.2 ObjectDrawerItem

Στην κλάση ObjectDrawerItem.java ορίζεται ένα αντικείμενο τύπου ObjectDrawerItem. Αυτού του τύπου αντικείμενα εισάγονται σαν στοιχεία στο navigation drawer. Τα χαρακτηριστικά (instance variables) αυτού του αντικειμένου είναι το όνομα και το εικονίδιο του.

Ακολουθεί ο κώδικας την κλάσης που δεν είναι κάτι άλλο από έναν απλό constructor.

```
public class ObjectDrawerItem {

    public int icon;
    public String name;

    // Constructor.
    public ObjectDrawerItem(int icon, String name) {

        this.icon = icon;
        this.name = name;
    }
}
```

3.3 DrawerItemCustomAdapter

Η DrawerItemCustomAdapter.java κλάση είναι ένας adapter για το navigation drawer. Όπως έχει αναφερθεί και παραπάνω το navigation drawer είναι ένα πάνελ που εμφανίζει τις κύριες επιλογές πλοήγησης της εφαρμογής. Αυτές η επιλογές εμφανίζονται σε μορφή λίστας. Αυτό που κάνει ο adapter, και κατ' επέκταση η κλάση, είναι να παίρνει τα αντικείμενα τύπου ObjectDrawerItem και να τα προσαρμόζει σε αυτή την λίστα. Συνεπώς, αυτό που εμφανίζεται στις επιλογές του navigation drawer είναι ένα εικονίδιο και ένα όνομα.

Ο κώδικας της κλάσης αυτής εμφανίζεται παρακάτω.

```
public class DrawerItemCustomAdapter extends ArrayAdapter<ObjectDrawerItem> {

    Context mContext;
    int layoutResourceId;
    ObjectDrawerItem data[] = null;

    public DrawerItemCustomAdapter(Context mContext, int layoutResourceId,
ObjectDrawerItem[] data) {

        super(mContext, layoutResourceId, data);
        this.layoutResourceId = layoutResourceId;
        this.mContext = mContext;
        this.data = data;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {

        View listItem = convertView;

        LayoutInflater inflater = ((Activity) mContext).getLayoutInflater();
        listItem = inflater.inflate(layoutResourceId, parent, false);

        ImageView imageViewIcon = (ImageView)
listItem.findViewById(R.id.imageViewIcon);
        TextView textViewName = (TextView)
listItem.findViewById(R.id.textViewName);

        ObjectDrawerItem folder = data[position];


        imageViewIcon.setImageResource(folder.icon);
        textViewName.setText(folder.name);

        return listItem;
    }

}
```

Κεφάλαιο 4

4 Η βάση δεδομένων της εφαρμογής

Για την καλύτερη κατανόηση των λειτουργιών της εφαρμογής, στο παρόν κεφάλαιο θα αναλυθεί η βάση δεδομένων που χρησιμοποιήθηκε.

Για την εγγραφή, το διάβασμα, την διαγραφή, και γενικώς τη διαχείριση των δεδομένων τις εφαρμογής γίνεται χρήση της SQLite Database βάση δεδομένων. Η SQLite είναι μία ανοιχτού κώδικα SQL βάση δεδομένων που αποθηκεύει δεδομένα σε ένα αρχείο κειμένου σε μια συσκευή. Το Android έρχεται με ενσωματωμένη εφαρμογή της SQLite βάσης δεδομένων.

Η SQLite υποστηρίζει όλα τα σχετικά με βάσεις δεδομένων χαρακτηριστικά. Για να αποκτηθεί πρόσβαση σε αυτή την βάση δεδομένων δεν χρειάζεται να δημιουργηθεί κάποιου είδους πρόσβαση. Η δημιουργία της βασίζεται πάνω στην γλώσσα SQL.

Για την εν λόγω εφαρμογή, η υλοποίηση της βάσης γίνεται στο αρχείο MyDBHandler.java, όπου δημιουργούνται τέσσερις διαφορετικοί πίνακες, ένας για κάθε λειτουργία της εφαρμογής. Στη συνέχεια παρουσιάζουμε την υλοποίηση της βάσης για κάθε μία λειτουργία ξεχωριστά.

4.1 Ορισμός της βάσης για την λειτουργία Calendar

4.1.1 Δημιουργία πίνακα

Ο πίνακας για τα δεδομένα της λειτουργίας Calendar δημιουργείται στην μέθοδο onCreate() και έχει πέντε στήλες. Το όνομα του πίνακα είναι “events” και τα δεδομένα που αποθηκεύει στις στήλες του είναι το αναγνωριστικό id του γεγονότος, το όνομα του γεγονότος, η ώρα έναρξης και λήξης σε milliseconds, και το χρώμα του γεγονότος που προσδιορίζει το είδος του.

```
String create_table_events = "CREATE TABLE " + TABLE_EVENTS + "(" +
        COLUMN_EVENTID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
        COLUMN_EVENTTITLE + " TEXT, " +
        COLUMN_STARTTIME + " LONG, " +
        COLUMN_ENDTIME + " LONG, " +
        COLUMN_DESCRIPTION + " INTEGER " +
        ")";
db.execSQL(create_table_events);
```

4.1.2 Προσθήκη και διαγραφή γεγονότων

Για την προσθήκη και την διαγραφή γεγονότων στον πίνακα της βάσης δεδομένων έχουν οριστεί δύο μέθοδοι. Η addEvent(Event event) παίρνει τα χαρακτηριστικά (τίτλος, ώρα έναρξης, ώρα λήξης, χρώμα) της παραμέτρου event και δημιουργεί μία καινούργια εγγραφή στον πίνακα.

```
public void addEvent(Event event){
    ContentValues values = new ContentValues();
    values.put(COLUMN_EVENTTITLE, event.getTitle());
    values.put(COLUMN_STARTTIME, event.getStartDate());
    values.put(COLUMN_ENDTIME, event.getEndDate());
    values.put(COLUMN_DESCRIPTION, event.getEventColor());
    SQLiteDatabase db = getWritableDatabase();
    db.insert(TABLE_EVENTS, null, values);
    db.close();
}
```

Η deleteEvent(String title, long starttime, long endtime) από την άλλη, διαγράφει την εγγραφή από τον πίνακα που έχει αυτά τα στοιχεία στις αντίστοιχες στήλες του.

```
public void deleteEvent(String eventtitle, long starttime, long endtime) {
    SQLiteDatabase db = getWritableDatabase();
    db.execSQL("DELETE FROM " + TABLE_EVENTS + " WHERE " +
            COLUMN_EVENTTITLE + "=\"" + eventtitle + "\"" + " AND " +
            COLUMN_STARTTIME + "=" + starttime + " AND " +
            COLUMN_ENDTIME + "=" + endtime + ";");
}
```

4.1.3 Διάβασμα γεγονότων από την βάση

Για το διάβασμα γεγονότων από τον πίνακα της βάσης έχουν οριστεί τέσσερις μέθοδοι. Για το διάβασμα όλων των τίτλων που περιέχει ο πίνακας έχει χρησιμοποιηθεί η μέθοδος eventTitleRetrieving(). Αυτή η μέθοδος επιστρέφει μία λίστα όπου στην κάθε γραμμή της περιέχει έναν τίτλο από τα γεγονότα που υπάρχουν στον πίνακα.

```
public List<String> eventTitleRetrieving(){
    List<String> dbString = new ArrayList<String>();
    SQLiteDatabase db = getWritableDatabase();
    String query = "SELECT * FROM " + TABLE_EVENTS + " WHERE 1";

    //Cursor points to a location in your results
    Cursor c = db.rawQuery(query, null);
    //Move to the first row in your results
    c.moveToFirst();

    //Position after the last row means the end of the results
    while (!c.isAfterLast()) {
        if (c.getString(c.getColumnIndex("eventtitle")) != null) {
            dbString.add(c.getString(c.getColumnIndex("eventtitle")));

        }
        c.moveToNext();
    }
    db.close();
    return dbString;
}
```

Για το διάβασμα των χρόνων, καθώς και του χρώματος των γεγονότων που υπάρχουν στον πίνακα έχουν οριστεί πανομοιότυπες μέθοδοι με αυτή που περιγράφτηκε παραπάνω.

```
public List<Long> startTimeRetrieving(){
    List<Long> dbString = new ArrayList<Long>();
    SQLiteDatabase db = getWritableDatabase();
    String query = "SELECT * FROM " + TABLE_EVENTS + " WHERE 1";

    //Cursor points to a location in your results
    Cursor c = db.rawQuery(query, null);
    //Move to the first row in your results
    c.moveToFirst();

    //Position after the last row means the end of the results
    while (!c.isAfterLast()) {
        if (c.getLong(c.getColumnIndex("starttime")) != 0) {
            dbString.add(c.getLong(c.getColumnIndex("starttime")));

        }
        c.moveToNext();
    }
    db.close();
    return dbString;
}



public List<Long> endTimeRetrieving(){
    List<Long> dbString = new ArrayList<Long>();
    SQLiteDatabase db = getWritableDatabase();
    String query = "SELECT * FROM " + TABLE_EVENTS + " WHERE 1";

    //Cursor points to a location in your results
    Cursor c = db.rawQuery(query, null);
    //Move to the first row in your results
    c.moveToFirst();

    //Position after the last row means the end of the results
    while (!c.isAfterLast()) {
        if (c.getLong(c.getColumnIndex("endtime")) != 0) {
            dbString.add(c.getLong(c.getColumnIndex("endtime")));

        }
        c.moveToNext();
    }
    db.close();
    return dbString;
}
```

```
public List<Integer> ColorRetrieving(){
    List<Integer> eventColor = new ArrayList<Integer>();
    SQLiteDatabase db = getWritableDatabase();
    String query = "SELECT * FROM " + TABLE_EVENTS + " WHERE 1";

    //Cursor points to a location in your results
    Cursor c = db.rawQuery(query, null);
    //Move to the first row in your results
    c.moveToFirst();

    //Position after the last row means the end of the results
    while (!c.isAfterLast()) {
        if (c.getInt(c.getColumnIndex("eventcolor")) != 0) {
            eventColor.add(c.getInt(c.getColumnIndex("eventcolor")));

        }
        c.moveToNext();
    }
    db.close();
    return eventColor;
}
```

4.2 Ορισμός της βάσης για την λειτουργία RememberToDo

4.2.1 Δημιουργία πίνακα

Όπως και στην λειτουργία Calendar, ο πίνακας για τα δεδομένα της λειτουργίας αυτής δημιουργείται στην μέθοδο onCreate(). Το όνομα που έχει επιλεγεί για αυτό τον πίνακα είναι “tasks” και έχει τρεις στήλες. Σε αυτές τις στήλες αποθηκεύονται το χαρακτηριστικό id του task, το όνομα του και μια τιμή για το αν αυτό το task έχει οριστεί σαν checked ή όχι.

```
String query = "CREATE TABLE " + TABLE_TASKS + "(" +
        COLUMN_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
        COLUMN_TASKNAME + " TEXT, " +
        COLUMN_CHECKED + " INTEGER " +
        ")";
db.execSQL(query);
```

4.2.2 Προσθήκη και διαγραφή εργασιών

Για την προσθήκη μιας εγγραφής στον πίνακα “tasks” έχει δημιουργηθεί η μέθοδος addTask(ObjectListTask task) η οποία παίρνει το όνομα και την κατάσταση της παραμέτρου task και τα προσθέτει σαν μια καινούργια εγγραφή στον πίνακα.

```
public void addTask(ObjectListTask task){
    ContentValues values = new ContentValues();
    values.put(COLUMN_TASKNAME, task.getName());
    int result;
    if (task.isChecked()){
        result = 1;
    }
    else {
        result = 0;
    }
    values.put(COLUMN_CHECKED, result);
    SQLiteDatabase db = getWritableDatabase();
    db.insert(TABLE_TASKS, null, values);
    db.close();
}
```

Αντιθέτως, για την διαγραφή ενός task από τον πίνακα έχουν οριστεί δύο μέθοδοι. Η μέθοδος deleteTask(String name) διαγράφει την εγγραφή με όνομα αυτό της παραμέτρου, ενώ η μέθοδος deleteTask() διαγράφει όλες τις εγγραφές οι οποίες η κατάσταση τους έχει την τιμή 1 (checked).

```
public void deleteTask(String taskname){
    SQLiteDatabase db = getWritableDatabase();

    db.execSQL("DELETE FROM " + TABLE_TASKS + " WHERE " + COLUMN_TASKNAME + "=
\"" + taskname + "\";");
}
```

```
public void deleteTask(){
    SQLiteDatabase db = getWritableDatabase();

    db.execSQL("DELETE FROM " + TABLE_TASKS + " WHERE " + COLUMN_CHECKED + "="
+ 1 + ";");
}
```

4.2.3 Διάβασμα εργασιών από την βάση

Η διαδικασία που ακολουθείται είναι η ίδια με αυτή του διαβάσματος γεγονότων από την βάση στην παράγραφο 4.1.3. Τόσο για το όνομα όσο και για την κατάσταση της εργασίας, δημιουργείται μία λίστα, όπου στις γραμμές της έχει τα ονόματα και τις καταστάσεις αντίστοιχα όλων των εγγραφών στον πίνακα.

```
public List<String> databaseToString(){
    List<String> dbString = new ArrayList<String>();
    SQLiteDatabase db = getWritableDatabase();
    String query = "SELECT * FROM " + TABLE_TASKS + " WHERE 1";

    //Cursor points to a location in your results
    Cursor c = db.rawQuery(query, null);
    //Move to the first row in your results
    c.moveToFirst();

    //Position after the last row means the end of the results
    while (!c.isAfterLast()) {
        if (c.getString(c.getColumnIndex("taskname")) != null) {
            dbString.add(c.getString(c.getColumnIndex("taskname")));

        }
        c.moveToNext();
    }
    db.close();
    return dbString;
}
```

```
public List<Boolean> databaseIsChecked(){
    List<Boolean> dbChecked = new ArrayList<Boolean>();
    SQLiteDatabase db = getWritableDatabase();
    String query = "SELECT * FROM " + TABLE_TASKS + " WHERE 1";

    //Cursor points to a location in your results
    Cursor c = db.rawQuery(query, null);
    //Move to the first row in your results
    c.moveToFirst();

    //Position after the last row means the end of the results
    while (!c.isAfterLast()) {
        if (c.getString(c.getColumnIndex("checked")) != null) {
            int value = c.getInt(c.getColumnIndex("checked"));
            if (value == 1) {
                dbChecked.add(Boolean.TRUE);
            }
            else {
                dbChecked.add(Boolean.FALSE);
            }

        }
        c.moveToNext();
    }
    db.close();
    return dbChecked;
}
```

4.2.4 Επεξεργασία της κατάστασης μιας εργασίας

Η μόνη λειτουργία της μεθόδου updateChecked(ObjectListTask task) είναι να αλλάζει την τιμή της κατάστασης του task από 0 σε 1 (unchecked σε checked) ή από 1 σε 0 (checked σε unchecked).

```
public void updateChecked(ObjectListTask task){
    ContentValues values = new ContentValues();
    int result;
    if (task.isChecked()){
        result = 1;
    }
    else {
        result = 0;
    }
    values.put(COLUMN_CHECKED, result);
    SQLiteDatabase db = getWritableDatabase();
    db.update(TABLE_TASKS, values, COLUMN_TASKNAME + "=\"" + task.getName() +
"\"", null);
    db.close();
}
```

4.3 Ορισμός της βάσης για την λειτουργία Settings

4.3.1 Δημιουργία πίνακα

Για την λειτουργία Settings δημιουργούνται δύο πίνακες, όπου ο καθένας περιέχει μία μόνο εγγραφή. Αυτή η εγγραφή αρχικοποιείται με την δημιουργία του πίνακα. Αυτοί οι δύο πίνακες ορίζονται, όπως και όλοι οι άλλοι, στην μέθοδο onCreate(). Τα ονόματα των πινάκων είναι scheduleSettings και anniversarySettings, όπου ο καθένας έχει από τρεις στήλες. Και στους δύο πίνακες η εγγραφή τους περιέχει το αναγνωριστικό id, την ώρα έναρξης σε milliseconds και την κατάσταση (checked/unchecked).

```
//The TABLE_SCHEDULE_SETTINGS has only one row
String create_table_sch_settings = "CREATE TABLE " + TABLE_SCHEDULE_SETTINGS +
"(" +
        COLUMN_SCHEDULE_SETTINGS_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
        COLUMN_SCHEDULE_TIME + " LONG, " +
        COLUMN_SCHEDULE_STATE + " INTEGER " +
        ")";
db.execSQL(create_table_sch_settings);
```

```
//Initialization of the TABLE_SCHEDULE_SETTINGS
ContentValues values = new ContentValues();
Calendar calendar = Calendar.getInstance();
calendar.set(Calendar.HOUR_OF_DAY,8);
calendar.set(Calendar.MINUTE,0);
calendar.set(Calendar.SECOND, 0);
values.put(COLUMN_SCHEDULE_TIME, calendar.getTimeInMillis());
values.put(COLUMN_SCHEDULE_STATE, 2);
//db = getWritableDatabase();
db.insert(TABLE_SCHEDULE_SETTINGS, null, values);
```

```
//The TABLE_ANNIVERSARY_SETTINGS has only one row
String create_table_ann_settings = "CREATE TABLE " + TABLE_ANNIVERSARY_SETTINGS
+ "(" +
        COLUMN_ANNIVERSARY_SETTINGS_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, "
+
        COLUMN_ANNIVERSARY_TIME + " LONG, " +
        COLUMN_ANNIVERSARY_STATE + " INTEGER " +
        ")";
db.execSQL(create_table_ann_settings);
```

```
//Initialization of the TABLE_ANNIVERSARY_TABLE
ContentValues value = new ContentValues();
Calendar cal = Calendar.getInstance();
cal.set(Calendar.HOUR_OF_DAY,8);
cal.set(Calendar.MINUTE,0);
cal.set(Calendar.SECOND, 0);
value.put(COLUMN_ANNIVERSARY_TIME, cal.getTimeInMillis());
value.put(COLUMN_ANNIVERSARY_STATE, 2);
//db = getWritableDatabase();
db.insert(TABLE_ANNIVERSARY_SETTINGS, null, value);
```

4.3.2 Ενημέρωση δεδομένων στους πίνακες scheduleSettings και anniversarySettings

Στους πίνακες scheduleSettings και anniversarySettings πρέπει να υπάρχει μόνο μία εγγραφή, για το λόγο αυτό δεν γίνεται προσθήκη κάποιας καινούργιας εγγραφής, παρά μόνο ενημέρωση της υπάρχουσας. Για τον πίνακα scheduleSettings έχουν οριστεί οι μέθοδοι setScheduleTime() και setScheduleState(), ενώ για τον πίνακα anniversarySettings έχουν οριστεί οι setAnniversaryTime() και setAnniversaryState() αντίστοιχα.

```
public void setScheduleTime(Long time){
    ContentValues values = new ContentValues();
    values.put(COLUMN_SCHEDULE_TIME, time);
    SQLiteDatabase db = getWritableDatabase();
    db.update(TABLE_SCHEDULE_SETTINGS, values, COLUMN_SCHEDULE_SETTINGS_ID +
"=" + 1, null);
    db.close();
}

public void setScheduleState(int state) {
    ContentValues values = new ContentValues();
    values.put(COLUMN_SCHEDULE_STATE, state);
    SQLiteDatabase db = getWritableDatabase();
    db.update(TABLE_SCHEDULE_SETTINGS, values, COLUMN_SCHEDULE_SETTINGS_ID +
"=" + 1, null);
    db.close();
}

public void setAnniversaryTime(Long time){
    ContentValues values = new ContentValues();
    values.put(COLUMN_ANNIVERSARY_TIME, time);
    SQLiteDatabase db = getWritableDatabase();
    db.update(TABLE_ANNIVERSARY_SETTINGS, values,
COLUMN_ANNIVERSARY_SETTINGS_ID + "=" + 1, null );
    db.close();
}

public void setAnniversaryState(int state) {
    ContentValues values = new ContentValues();
    values.put(COLUMN_ANNIVERSARY_STATE, state);
    SQLiteDatabase db = getWritableDatabase();
    db.update(TABLE_ANNIVERSARY_SETTINGS, values,
COLUMN_ANNIVERSARY_SETTINGS_ID + "=" + 1, null );
    db.close();
}
```

4.3.3 Διάβασμα δεδομένων από την βάση

Για την ανάγνωση από τους πίνακες scheduleSettings και anniversarySettings έχουν δημιουργηθεί ανάλογες μέθοδοι, με αυτές των άλλων πινάκων, που έχουν καλυφτεί παραπάνω στο κεφάλαιο αυτό.

```
public long timeScheduleRetrieving (){
    Long time = new Long(0);
    SQLiteDatabase db = getWritableDatabase();
    String query = "SELECT * FROM " + TABLE_SCHEDULE_SETTINGS + " WHERE 1";

    //Cursor points to a location in your results
    Cursor c = db.rawQuery(query, null);
    //Move to the first row in your results
    c.moveToFirst();

    //Position after the last row means the end of the results
    while (!c.isAfterLast()) {
        if (c.getLong(c.getColumnIndex("scheduleTime")) != 0) {
            time = (c.getLong(c.getColumnIndex("scheduleTime")));

        }
        c.moveToNext();
    }
    db.close();
    return time;
}



public int stateScheduleRetrieving (){
    int state = new Integer(-1);
    SQLiteDatabase db = getWritableDatabase();
    String query = "SELECT * FROM " + TABLE_SCHEDULE_SETTINGS + " WHERE 1";

    //Cursor points to a location in your results
    Cursor c = db.rawQuery(query, null);
    //Move to the first row in your results
    c.moveToFirst();

    //Position after the last row means the end of the results
    while (!c.isAfterLast()) {
        if (c.getInt(c.getColumnIndex("scheduleState")) != 0) {
            state = (c.getInt(c.getColumnIndex("scheduleState")));

        }
        c.moveToNext();
    }
    db.close();
    return state;
}
```

```
public long timeAnniversaryRetrieving (){
    Long time = new Long(0);
    SQLiteDatabase db = getWritableDatabase();
    String query = "SELECT * FROM " + TABLE_ANNIVERSARY_SETTINGS + " WHERE 1";

    //Cursor points to a location in your results
    Cursor c = db.rawQuery(query, null);
    //Move to the first row in your results
    c.moveToFirst();

    //Position after the last row means the end of the results
    while (!c.isAfterLast()) {
        if (c.getLong(c.getColumnIndex("anniversaryTime")) != 0) {
            time = (c.getLong(c.getColumnIndex("anniversaryTime")));

        }
        c.moveToNext();
    }
    db.close();
    return time;
}



public int stateAnniversaryRetrieving (){
    int state = new Integer(-1);
    SQLiteDatabase db = getWritableDatabase();
    String query = "SELECT * FROM " + TABLE_ANNIVERSARY_SETTINGS + " WHERE 1";

    //Cursor points to a location in your results
    Cursor c = db.rawQuery(query, null);
    //Move to the first row in your results
    c.moveToFirst();

    //Position after the last row means the end of the results
    while (!c.isAfterLast()) {
        if (c.getInt(c.getColumnIndex("anniversaryState")) != 0) {
            state = (c.getInt(c.getColumnIndex("anniversaryState")));

        }
        c.moveToNext();
    }
    db.close();
    return state;
}
```

Κεφάλαιο 5

5 Η λειτουργία Calendar

5.1 Η βιβλιοθήκη Telerik UI για Android

Πρόκειται για μια βιβλιοθήκη από εγγενή στοιχεία διεπαφής χρήστη, κατασκευασμένη από κάτω-προς-τα-πάνω, για να παρέχει γρήγορη φόρτωση, άριστες δυνατότητες σχεδίασης και κομψή διαδραστικότητα σε οποιαδήποτε συσκευή και μέγεθος οθόνης τρέχει το λειτουργικό Android. Η βιβλιοθήκη παρέχει γραφικά, καθώς και λειτουργικότητες για διαγράμματα, ημερολόγια, λίστες και πολλά άλλα.

Η παρούσα εφαρμογή χρησιμοποιεί αυτή το Telerik UI για να "δανειστεί" τα γραφικά του και λειτουργικότητες που παρέχονται για το ημερολόγιο.

Εικόνα 5-1

5.2 Περιγραφή της λειτουργίας

Όταν ο χρήστης επιλέξει από το navigation drawer την επιλογή Calendar τότε μεταβαίνει στο RadCalendarFragment. Αυτό που θα δει, είναι μία γραφική απεικόνιση ενός ημερολογίου.

Εικόνα 5-2

Το ημερολόγιο είναι σε μηνιαία μορφή, εμφανίζοντας τις ημέρες και τις εβδομάδες του μήνα όπου βρίσκεται. Με slide πάνω και κάτω, ο χρήστης μεταβαίνει στους διπλανούς μήνες. Κάνοντας ένα απλό zoom out το ημερολόγιο αλλάζει από μηνιαία σε ετήσια μορφή και πλέον είναι σε θέση να μεταβεί σε όποιον μήνα επιθυμεί. Επιπλέον με slide πάνω και κάτω μπορεί μεταβαίνει σε διπλανά έτη.

Εικόνα 5-3

Πατώντας το κουμπί menu και επιλέγοντας το “Add New Event”, ο χρήστης πηγαίνει στο NewEventFragment. Σε αυτό το fragment γίνεται η καταχώριση ενός νέου γεγονότος στη βάση δεδομένων. Του ζητείται να πληκτρολογήσει το όνομα του συμβάντος, την ημερομηνία και την ώρα έναρξης, την ημερομηνία και την ώρα λήξης, καθώς και το τι “είδος” είναι το συμβάν (appointment, anniversary, other). Σε περίπτωση που το γεγονός έχει επιλεγεί ως anniversary, η ώρα έναρξης-λήξης καταχωρούνται αυτόματα. Πατώντας το κουμπί “Save” αποθηκεύει το γεγονός στη βάση και επιστρέφει στο RadCalendarFragment όπου εμφανίζεται και πάλι το ημερολόγιο. Αυτόματα, αν το γεγονός έχει καταχωρηθεί ως appointment ή other, κατασκευάζεται ένα notification για το συμβάν στην ώρα έναρξης. Πατώντας το κουμπί “Cancel” γίνεται επιστροφή στο ημερολόγιο και όλη η διαδικασία ακυρώνεται.

Όταν υπάρχουν καταχωρημένα γεγονότα στη βάση δεδομένων, αυτά εμφανίζονται στο ημερολόγιο σαν χρωματιστά τετράγωνα, όπου το χρώμα προσδιορίζει το είδος του γεγονότος. Μπλε για appointment, κόκκινο για anniversary και πράσινο για other. Πατώντας πάνω σε μία ημερομηνία όπου υπάρχουν προγραμματισμένα γεγονότα, εμφανίζεται ένα dialog παράθυρο με όλα τα γεγονότα με τις ώρες έναρξης-λήξης (σε περίπτωση appointment ή other) που υπάρχουν στη βάση δεδομένων με αυτή την ημερομηνία. Σε περίπτωση που πρέπει να διαγραφεί ένα γεγονός από την βάση δεδομένων, πατώντας παρατεταμένα πάνω στο γεγονός του dialog παραθύρου, αυτό διαγράφεται.

Εικόνα 5-4

Παρακάτω γίνεται πιο λεπτομερής ανάλυση και παρουσίαση του κώδικα.

5.3 RadCalendarFragment

Για την εμφάνιση του ημερολογίου, βασικό ρόλο έχει η κλάση RadCalendarFragment.java. Στην αρχή αυτής της κλάσης, γίνεται import η βιβλιοθήκη, για την διαχείριση των γραφικών του ημερολογίου και των λειτουργιών των events. Στη μέθοδο onViewCreated(), γίνονται η δήλωση του ημερολογίου, το διάβασμα των αποθηκευμένων γεγονότων από τη βάση δεδομένων και η διαχείριση τους.

5.3.1 onCreateView() Method

Στο εσωτερικό αυτής της μεθόδου πραγματοποιείται η σύνδεση του fragment_radcalendar.xml, που περιγράφει τα γραφικά του ημερολογίου, με την κλάση μας.

Ακολουθεί ο κώδικας της onCreateView() και του fragment_radcalendar.xml

```
public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
    View rootView = inflater.inflate(R.layout.fragment_radcalendar, container,
false);
    return rootView;
}
```

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:calendar="http://schemas.android.com/apk/res-auto"
    android:id="@+id/fragment_radcalendar"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".RadCalendarFragment">

    <com.telerik.widget.calendar.RadCalendarView
        android:id="@+id/radcalendarView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>


</RelativeLayout>
```

5.3.2 onViewCreated() Method

Όπως αναφέρθηκε και παραπάνω, σε αυτή την μέθοδο γίνεται η σύνδεση των στοιχείων του fragment_radcalendar.xml αρχείου με τις μεταβλητές της κλάσης. Εν συνεχεία, τίθεται μία βελτιστοποίηση της

εμφάνισης του ημερολογίου, όπως είναι η αλλαγή του χρώματος του φόντου, το τι σχήμα θα εμφανίζεται στην επιλογή της ημερομηνίας, καθώς και σε ποιο σημείο θα εμφανίζεται ο αριθμός της ημερομηνίας μέσα στο τετράγωνο.

Όλα αυτά καθορίζονται με τον παρακάτω κώδικα.

```
//declaration of CalendarView
RadCalendarView calendarView = (RadCalendarView)
rootView.findViewById(R.id.radcalendarView);
eventInfo = (TextView) rootView.findViewById(R.id.event_info);
Calendar calendar = Calendar.getInstance();

//CalendarView optimization
calendarView.setSelectionMode(CalendarSelectionMode.Single);
calendarView.getAdapter().setDateTextPosition(CalendarElement.CENTER);
calendarView.getAdapter().setDateCellBackgroundColor(Color.WHITE, Color.WHITE);
calendarView.getAdapter().setSelectedCellBackgroundColor(Color.WHITE);

CellDecorator decorator = new CircularCellDecorator(calendarView);
decorator.setColor(Color.parseColor("#ed742c"));
decorator.setStrokeWidth(Util.getDimen(TypedValue.COMPLEX_UNIT_DIP, 2));
decorator.setScale(.75f);
decorator.setStroked(true);

calendarView.setCellDecorator(decorator);
calendarView.setDisplayDate(calendar.getTimeInMillis());
```

Έπειτα, τα γεγονότα φορτώνονται από την βάση δεδομένων. Δημιουργείται μία λίστα από events όπου σε κάθε εγγραφή της έχει και ένα event από την βάση δεδομένων. Αφού δημιουργηθεί η λίστα με τα events της βάσης, αυτά προσθέτονται στο ημερολόγιο.

```
//Retrieve the stored events of the database
dbHandler = new MyDBHandler(getActivity().getApplicationContext(), null, null,
1);
List<Event> mEvents = new ArrayList<Event>();
final List<String> eventTitle = dbHandler.eventTitleRetrieving();
final List<Long> eventStartTime = dbHandler.startTimeRetrieving();
final List<Long> eventEndTime = dbHandler.endTimeRetrieving();
final List<Integer> eventDescription = dbHandler.ColorRetrieving();

for (int i=0; i<eventTitle.size(); i++){
    Event event = new Event(eventTitle.get(i),eventStartTime.get(i),
eventEndTime.get(i));
    event.setEventColor(eventDescription.get(i));
    mEvents.add(event);
}

//put the events to the calendarView
calendarView.getEventAdapter().setEvents(mEvents);
```

Όταν ολοκληρωθεί αυτό το βήμα και τα events πλέον φαίνονται στο ημερολόγιο, συνέχεια έχει η διαχείριση των κελιών (ημερών) του ημερολογίου. Όταν ο χρήστης πατήσει πάνω σε μία ημερομηνία τότε κατασκευάζεται μία λίστα με τα events που υπάρχουν στη συγκεκριμένη ημερομηνία. Αυτή η λίστα εμφανίζεται στην οθόνη μέσω ενός dialog παραθύρου. Παρατεταμένη πίεση πάνω σε μία γραμμή της λίστας συνεπάγεται την διαγραφή του event από την βάση δεδομένων και την μη εμφάνιση του πλέον στο ημερολόγιο.

Εικόνα 5-5

Παρατίθεται ο κώδικας των όσων αναφέρθηκαν παραπάνω.

```
//code executed when a calendar's cell is clicked
calendarView.setOnCellClickListener(new RadCalendarView.OnCellClickListener() {
    @Override
    public void onCellClick(CalendarCell calendarCell) {
        if (!(calendarCell instanceof CalendarDayCell))
            return;

        CalendarDayCell calendarDayCell = (CalendarDayCell) calendarCell;

        //check if the cell(date) has events
        if (calendarDayCell.getEvents() != null &&
                calendarDayCell.getEvents().size() > 0) {

            listView = new ListView(getActivity());
            final List<Event> dailyEvents = calendarDayCell.getEvents();
            final ArrayList<String> mDailyEventsInfo = new ArrayList<String>();
            final ArrayAdapter<String> adapter = new ArrayAdapter<String>
(getActivity(),R.layout.listview_events_row,
                    R.id.event_info, mDailyEventsInfo);

            //for these events create a dialog box this each info
            for (int i=0; i<dailyEvents.size(); i++) {

                if (dailyEvents.get(i).getEventColor() == Color.RED) {
                    String eventInfo = String.format("Anniversary: %s",
                            dailyEvents.get(i).getTitle());
                    mDailyEventsInfo.add(eventInfo);
                }
                else {
                    String eventInfo = String.format("%tT - %tT: %s",
                            dailyEvents.get(i).getStartDate(),
                            dailyEvents.get(i).getEndDate(),
                            dailyEvents.get(i).getTitle());
                    mDailyEventsInfo.add(eventInfo);
                }
            }

            listView.setAdapter(adapter);
            //long press on the event info will delete it from the database
            listView.setOnItemLongClickListener(new
AdapterView.OnItemLongClickListener() {
                @Override
                public boolean onItemLongClick(AdapterView<?> parent, View
view, int position, long id) {
                    dbHandler.deleteEvent(dailyEvents.get(position).getTitle(),
                            dailyEvents.get(position).getStartDate(),
                            dailyEvents.get(position).getEndDate());
                    mDailyEventsInfo.remove(mDailyEventsInfo.get(position));
                    adapter.notifyDataSetChanged();
                    onViewCreated(rootView, savedInstanceState);
                    return true;

                }
            });

            //creating of the dialog box
            AlertDialog.Builder builder = new
AlertDialog.Builder(getActivity());
            builder.setCancelable(true);
            builder.setPositiveButton("OK", null);
            builder.setView(listView);
            AlertDialog dialog = builder.create();
            dialog.setCanceledOnTouchOutside(true);
            dialog.show();
        }
    }
});
```

5.3.3 onCreateOptionMenu() και onOptionsItemSelected() Methods

Τέλος, υπάρχουν και κάποιες μέθοδοι για την διαχείριση του κουμπιού menu.

Για να εμφανιστεί το OptionMenu στο fragment πρέπει να έχει οριστεί πρώτα στην μέθοδο onCreate() του. Αυτό γίνεται με την εντολή setHasOptionMenu(true);.

Στην μέθοδο onCreateOptionMenu() δημιουργείται η επιλογή που εμφανίζεται όταν πατηθεί το κουμπί menu.

```
public void onCreateOptionsMenu(Menu menu, MenuInflater inflater) {
    super.onCreateOptionsMenu(menu, inflater);
    getActivity().getMenuInflater().inflate(R.menu.menu_calendar, menu);
}
```

Το αρχείο menu_calendar.xml ορίζει το στοιχείο του OptionMenu όπως περιγράφει ο παρακάτω κώδικας.

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
tools:context=".RadCalendarFragment">
    <item android:id="@+id/action_add_event"
android:title="@string/action_add_event"
        android:orderInCategory="100" app:showAsAction="never" />
</menu>
```

Στη μέθοδο onOptionsItemSelected() γίνεται ένας έλεγχος για το αν έχει επιλεγεί το στοιχείο με αναγνωριστικό το action_add_event όπως ορίζεται στον κώδικα που βρίσκεται από πάνω. Αν το στοιχείο έχει επιλεγεί, τότε δημιουργείται το NewEventFragment.

```
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if (id == R.id.action_add_event) {
        NewEventFragment newEvent = new NewEventFragment();
        FragmentManager fragmentManager =
getActivity().getSupportFragmentManager();
        fragmentManager.beginTransaction().replace(R.id.content_frame,
newEvent).addToBackStack(null).commit();
    }
    return true;
}
```

5.4 NewEventFragment

Σε αυτή την κλάση γίνεται η εισαγωγή ενός καινούργιο γεγονότος στη βάση δεδομένων και κατ' επέκταση και στο ημερολόγιο. Ο χρήστης καλείται να εισάγει τα απαραίτητα χαρακτηριστικά για το καινούργιο event όπως φαίνονται στην Εικόνα 5-3.

Ο constructor που παρέχει η βιβλιοθήκη για τα αντικείμενα event υποστηρίζει πως ένα event θα πρέπει να χαρακτηρίζεται από το όνομα του, τον χρόνο έναρξης (έτος, μήνας, μέρα, ώρα, λεπτά) σε milliseconds και τον χρόνο λήξης (έτος, μήνας, μέρα, ώρα, λεπτά) σε milliseconds. Ένα χαρακτηριστικό που μπορεί να έχει ένα event, αλλά δεν είναι υποχρεωτικό, είναι το χρώμα του. Στην εφαρμογή, το χρώμα χρησιμοποιείται για να χαρακτηρίσει το είδος του event (appointment, anniversary, other). Κάτι που έχει πολύ μεγάλη σημασία είναι ότι ο constructor του event είναι έτσι ορισμένος ώστε να μην επιτρέπει τον χρόνο έναρξης να είναι μεγαλύτερος του χρόνου λήξης.

Στην αρχή της κλάσης γίνεται import η βιβλιοθήκη Telerik UI που διαχειρίζεται τα events. Στην μέθοδο onViewCreated() γίνεται η διαχείριση τον κουμπιών, και οι έλεγχοι για το αν έχει γίνει σωστή εισαγωγή των δεδομένων. Στο τέλος αυτής της κλάσης ορίζεται μία μέθοδος για την δημιουργία notification.

5.4.1 onCreateView() Method

Αυτή η μέθοδος συνδέει το αρχείο που περιέχει την γραφική απεικόνιση του fragment, fragment_new_event.xml, με την κλάση ώστε να υλοποιηθούν οι λειτουργίες.

Ακολουθεί ο κώδικας της μεθόδου και του fragment_new_event.xml αρχείου.

```
public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
    View rootView = inflater.inflate(R.layout.fragment_new_event, container,
false);
    return rootView;
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:id="@+id/fragment_new_event"
    tools:context=".NewEventFragment"
    android:weightSum="10">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:padding="10dp"
        android:text="@string/add_title"
        android:id="@+id/add_title"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:padding="10dp"
        android:text="@string/event_title"
        android:id="@+id/event_name_text"
        android:layout_below="@+id/add_title"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="10dp"
        android:id="@+id/event_name_input"
        android:hint="@string/event_title_input"
        android:layout_below="@+id/event_name_text"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:padding="10dp"
        android:text="@string/start_date"
        android:id="@+id/start_date_text"
        android:layout_toRightOf="@+id/start_date_input"
        android:layout_toEndOf="@+id/start_date_input"
        android:layout_alignBottom="@+id/start_date_input"
        android:gravity="bottom" />
```

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:id="@+id/start_date_input"
    android:text="@string/start_date"
    android:layout_below="@+id/event_name_input"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:padding="10dp"
    android:text="@string/start_time"
    android:id="@+id/start_time_text"
    android:layout_toRightOf="@+id/start_time_input"
    android:layout_toEndOf="@+id/start_time_input"
    android:layout_alignBottom="@+id/start_time_input"
    android:gravity="bottom" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:id="@+id/start_time_input"
    android:text="@string/start_time"
    android:layout_below="@+id/end_date_input"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:padding="10dp"
    android:text="@string/end_date"
    android:id="@+id/end_date_text"
    android:layout_toRightOf="@+id/end_date_input"
    android:layout_toEndOf="@+id/end_date_input"
    android:layout_alignBottom="@+id/end_date_input"
    android:gravity="bottom" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:id="@+id/end_date_input"
    android:text="@string/end_date"
    android:layout_below="@+id/start_date_input"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:padding="10dp"
    android:text="@string/end_time"
    android:id="@+id/end_time_text"
    android:layout_alignBottom="@+id/end_time_input"
    android:layout_toRightOf="@+id/end_time_input"
    android:layout_toEndOf="@+id/end_time_input"
    android:gravity="bottom" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:id="@+id/end_time_input"
    android:text="@string/end_time"
    android:layout_below="@+id/start_time_input"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:padding="10dp"
    android:text="@string/event_description"
    android:id="@+id/description_text"
    android:layout_below="@+id/end_time_input"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_marginTop="20dp" />

<RadioGroup
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_below="@+id/description_text"
    android:id="@+id/event_description_input"
    android:orientation="horizontal">

    <RadioButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/appointment_button"
        android:text="@string/appointment"
        android:checked="false"
        android:layout_marginRight="10dp"
        android:singleLine="true"
        android:buttonTint="@color/blue" />
```

```
    <RadioButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/anniversary_button"
            android:text="@string/anniversary"
            android:layout_marginRight="10dp"
            android:buttonTint="@color/red" />

        <RadioButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/other_button"
            android:text="@string/other"
            android:layout_marginRight="10dp"
            android:buttonTint="@color/lightgreen" />

    </RadioGroup>
    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:weightSum="1"
        android:layout_below="@+id/description_text">

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/cancel_button"
            android:id="@+id/cancel_button"
            android:layout_weight="0.50"
            android:layout_gravity="bottom" />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/save_button"
            android:id="@+id/save_button"
            android:layout_weight="0.50"
            android:layout_gravity="bottom" />
    </LinearLayout>
</RelativeLayout>
```

5.4.2 onViewCreated() Method

Η μέθοδος onViewCreated() εκτελεί τις βασικές λειτουργίες του fragment. Αρχικά γίνεται η σύνδεση των στοιχείων του fragment_new_event.xml με τις μεταβλητές της κλάσης. Στη συνέχεια ορίζεται σε μια μεταβλητή ένα “στιγμιότυπο” της βάσης δεδομένων και

δημιουργείται ένα event με όνομα το όνομα που έχει πληκτρολογήσει ο χρήστης, χρόνο έναρξης το μηδέν, και χρόνο λήξης το 1.

```
event_name_input = (EditText) rootView.findViewById(R.id.event_name_input);
start_date_text = (TextView) rootView.findViewById(R.id.start_date_text);
start_time_text = (TextView) rootView.findViewById(R.id.start_time_text);
start_date_input = (Button) rootView.findViewById(R.id.start_date_input);
start_time_input = (Button) rootView.findViewById(R.id.start_time_input);
end_date_text = (TextView) rootView.findViewById(R.id.end_date_text);
end_time_text = (TextView) rootView.findViewById(R.id.end_time_text);
end_date_input = (Button) rootView.findViewById(R.id.end_date_input);
end_time_input = (Button) rootView.findViewById(R.id.end_time_input);
event_description_input = (RadioGroup)
rootView.findViewById(R.id.event_description_input);
appointment_button = (RadioButton)
rootView.findViewById(R.id.appointment_button);
anniversary_button = (RadioButton)
rootView.findViewById(R.id.anniversary_button);
other_button = (RadioButton) rootView.findViewById(R.id.other_button);
cancel_button = (Button) rootView.findViewById(R.id.cancel_button);
save_button = (Button) rootView.findViewById(R.id.save_button);

//create an instance of our database
dbHandler = new MyDBHandler(getActivity().getApplicationContext(), null, null,
1);
//create an instance of calendar
final Calendar calendar = Calendar.getInstance();
//initialize an event object Event(title,startTime,endTime),
//startTime should always be before endTime
final Event event = new Event(event_name_input.getText().toString(), 0, 1);
```

Αυτό που ακολουθεί στη μέθοδο έχει να κάνει με την διασύνδεση με τον χρήστη. Όποτε ο χρήστης πατήσει κάποιο κουμπί τότε εκτελείται και το κατάλληλο κομμάτι κώδικα. Οποιαδήποτε εισαγωγή δεδομένων με το πάτημα ενός κουμπιού, συνεπάγεται αλλαγή και στο event που έχει οριστεί παραπάνω. Έτσι ο χρήστης σιγά σιγά “χτίζει” το event που έχει σκοπό να αποθηκεύσει στη βάση δεδομένων.

5.4.2.1 Start date button

Με το πάτημα του κουμπιού “Start Date” καλείται η κλάση SelectDateFragment.java. Αυτή η κλάση δεν είναι τίποτα άλλο από ένα dialog παράθυρο για την επιλογή ημερομηνίας. Όταν ο χρήστης επιλέξει ημερομηνία, τότε αυτή αντικαθιστά τον προηγούμενο χρόνο έναρξης του event. Εκείνο που πρέπει να επισημανθεί είναι ότι ο χρόνος έναρξης του event, σε αυτό το σημείο, έχει έτος, μήνα και μέρα. Η ώρα και τα λεπτά θα συμπληρωθούν παρακάτω. Επίσης, αντικαθιστάται και ο

χρόνος λήξης με την τιμή “χρόνος έναρξης + 1”, ώστε να αποφευχθεί το σφάλμα στο οποίο ο χρόνος έναρξης είναι μεγαλύτερος από τον χρόνο λήξης.

```
start_date_input.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        SelectDateFragment date = new SelectDateFragment();

        // set date of today
        Calendar calender = Calendar.getInstance();
        Bundle args = new Bundle();
        args.putInt("year", calender.get(Calendar.YEAR));
        args.putInt("month", calender.get(Calendar.MONTH));
        args.putInt("day", calender.get(Calendar.DAY_OF_MONTH));
        date.setArguments(args);
        date.setCallBack(ondate);
        date.show(getChildFragmentManager(), "Date Picker");
    }
    DatePickerDialog.OnDateSetListener ondate = new
DatePickerDialog.OnDateSetListener() {
        @Override
        public void onDateSet(DatePicker view, int year, int monthOfYear, int
dayOfMonth) {
            int month = monthOfYear+1;
            start_date_text.setText(dayOfMonth + "/" + month + "/" + year);
            try {
                //put the chosen values at the calendar instance and set them
at the event
                //only the Year, Month, Day are chosen, the time remains the
same as
                //had initialized
                calendar.set(Calendar.YEAR, year);
                calendar.set(Calendar.MONTH, monthOfYear);
                calendar.set(Calendar.DAY_OF_MONTH, dayOfMonth);
                long eventStart = calendar.getTimeInMillis();
                event.setEndDate(eventStart+1);  //throws exception if the
startTime > endTime
                event.setStartDate(eventStart);
            }
            catch (Exception e) {
                //error message displayed if something gone bad
                start_date_text.setText("Start Date");
                Toast.makeText(
                        getActivity().getApplicationContext(),
                        "endDate should be after startDate",
                        Toast.LENGTH_LONG).show();
            }
        }
    };
});
```

Παρακάτω, ο κώδικας της SelectDateFragment.java κλάσης.

```
public class SelectDateFragment extends DialogFragment {

    private static final String TAG = "DatePickerFragment";
    DatePickerDialog.OnDateSetListener ondateSet;

    private int year;
    private int month;
    private int day;

    public SelectDateFragment() {}

    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        return new DatePickerDialog(getActivity(), ondateSet, year, month,
day);
    }

    @Override
    public void setArguments(Bundle args) {
        super.setArguments(args);
        year = args.getInt("year");
        month = args.getInt("month");
        day = args.getInt("day");
    }

    public void setCallBack(DatePickerDialog.OnDateSetListener ondate) {
        ondateSet = ondate;
    }

}
```

5.4.2.2 End date button

Ακολουθείται η ίδια λογική όπως και για το Start date button. Για το λόγο αυτό παρατίθεται μόνο ο κώδικας που χειρίζεται το πάτημα του κουμπιού.

```
end_date_input.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        //showDatePicker();
        SelectDateFragment date = new SelectDateFragment();

        // set date of today
        //Calendar calender = Calendar.getInstance();
        Bundle args = new Bundle();
        args.putInt("year", calendar.get(Calendar.YEAR));
        args.putInt("month", calendar.get(Calendar.MONTH));
        args.putInt("day", calendar.get(Calendar.DAY_OF_MONTH));
        date.setArguments(args);
        date.setCallBack(ondate);
        date.show(getChildFragmentManager(), "Date Picker");
    }
    DatePickerDialog.OnDateSetListener ondate = new
DatePickerDialog.OnDateSetListener() {
        @Override
        public void onDateSet(DatePicker view, int year, int monthOfYear, int
dayOfMonth) {
            int month = monthOfYear+1;
            end_date_text.setText(dayOfMonth + "/" + month + "/" + year);
            try {
                //put the chosen values at the calendar instance and set them
at the event
                //only the Year, Month, Day are chosen, the time remains the
same as
                //had initialize
                calendar.setTimeInMillis(event.getEndDate());
                calendar.set(Calendar.YEAR, year);
                calendar.set(Calendar.MONTH, monthOfYear);
                calendar.set(Calendar.DAY_OF_MONTH, dayOfMonth);
                long eventEnd = calendar.getTimeInMillis();
                event.setEndDate(eventEnd);
            } catch (Exception e) {
                //error message displayed if something gone bad
                end_date_text.setText("End Date");
                Toast.makeText(
                    getActivity().getApplicationContext(),
                    "endDate should be after startDate",
                    Toast.LENGTH_LONG).show();
            }
        }
    };
});
```

5.4.2.3 Start time button

Στην παράγραφο 5.4.2.1 όπου περιγράφτηκε το Start date button αναφέρθηκε πως από τον χρόνο έναρξης του event αντικαταστάθηκε μόνο το έτος, ο μήνας και η μέρα. Όταν ο χρήστης πατήσει το κουμπί Start Time και επιλέξει ώρα, στον χρόνο έναρξης προστίθεται πλέον και η ώρα με τα λεπτά. Το event σε αυτό το σημείο έχει ολοκληρωμένο τον

χρόνο έναρξης, δηλαδή έχουν οριστεί το έτος, ο μήνας, η μέρα, η ώρα και τα λεπτά. Η δομή του κώδικα είναι ίδια με αυτή του Start date button, με την μόνη διαφορά ότι τώρα καλείται η κλάση SelectTimeFragment.java που εμφανίζει ένα dialog παράθυρο για την επιλογή ώρας και λεπτών.

```
start_time_input.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        SelectTimeFragment date = new SelectTimeFragment();

        // set date of today
        Bundle args = new Bundle();
        args.putInt("hour", calendar.get(Calendar.HOUR_OF_DAY));
        args.putInt("minutes", calendar.get(Calendar.MINUTE));
        date.setArguments(args);
        date.setCallBack(ondate);
        date.show(getChildFragmentManager(), "Time Picker");
    }
    TimePickerDialog.OnTimeSetListener ondate = new
TimePickerDialog.OnTimeSetListener() {
        @Override
        public void onTimeSet(TimePicker view, int hourOfDay, int minute) {
            String time;
            if (minute<10){
                time = (hourOfDay + ":0" + minute);
            }
            else{
                time = (hourOfDay + ":" + minute);
            }
            start_time_text.setText(time);
            //get the chosen starTime from before and this time we set only the
Hour, Minute
            calendar.setTimeInMillis(event.getStartDate());
            try {
                //put the chosen values at the calendar instance and set them
at the event
                calendar.set(Calendar.HOUR_OF_DAY, hourOfDay);
                calendar.set(Calendar.MINUTE, minute);
                long eventStart = calendar.getTimeInMillis();
                event.setEndDate(eventStart+1);  //throws exception if the
startTime > endTime
                event.setStartDate(eventStart);
            }
            catch (Exception e){
                //error message displayed if something gone bad
                end_date_text.setText("End Time");
                Toast.makeText(
                        getActivity().getApplicationContext(),
                        "endDate should be after startDate",
                        Toast.LENGTH_LONG).show();
            }

        }
    };
});
```

Στη συνέχεια φαίνεται ο κώδικας της SelectTimeFragment.java

```
public class SelectTimeFragment extends DialogFragment{

    private static final String TAG = "DatePickerFragment";
    TimePickerDialog.OnTimeSetListener ontimeSet;

    private int hour;
    private int minutes;

    public SelectTimeFragment() {}

    @NonNull
    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        return new TimePickerDialog(getActivity(), ontimeSet, hour, minutes,
true);
    }

    @Override
    public void setArguments(Bundle args) {
        super.setArguments(args);
        hour = args.getInt("hour");
        minutes = args.getInt("minutes");
    }

    public void setCallBack(TimePickerDialog.OnTimeSetListener ontime) {
        ontimeSet = ontime;
    }
}
```

5.4.2.4 End time button

Όπως και παραπάνω, έτσι και εδώ, επιλέγεται η ώρα για τον χρόνο λήξης του event.

```
end_time_input.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        SelectTimeFragment date = new SelectTimeFragment();

        // set date of today
        Calendar calender = Calendar.getInstance();
        Bundle args = new Bundle();
        args.putInt("hour", calender.get(Calendar.HOUR_OF_DAY));
        args.putInt("minutes", calender.get(Calendar.MINUTE));
        date.setArguments(args);
        date.setCallBack(ondate);
        date.show(getChildFragmentManager(), "Time Picker");
    }
    TimePickerDialog.OnTimeSetListener ondate = new
TimePickerDialog.OnTimeSetListener() {
        @Override
        public void onTimeSet(TimePicker view, int hourOfDay, int minute) {
            String time;
            if (minute<10){
                time = (hourOfDay + ":0" + minute);
            }
            else{
                time = (hourOfDay + ":" + minute);
            }
            end_time_text.setText(time);
            //get the chosen endTime from before and this time we set only the
Hour, Minute
            calendar.setTimeInMillis(event.getEndDate());
            try{
                //put the chosen values at the calendar instance and set them
at the event
                calendar.set(Calendar.HOUR_OF_DAY, hourOfDay);
                calendar.set(Calendar.MINUTE, minute);
                long eventEnd = calendar.getTimeInMillis();
                event.setEndDate(eventEnd);
            } catch (Exception e) {
                //error message displayed if something gone bad
                end_time_text.setText("End Time");
                Toast.makeText(
                        getActivity().getApplicationContext(),
                        "endDate should be after startDate",
                        Toast.LENGTH_LONG).show();
            }
        }
    };
});
```

5.4.2.5 Description buttons

Με τα description buttons ο χρήστης καλείται να επιλέξει το είδος του event που θέλει να καταχωρίσει. Επιλέγοντας το κουμπί με το όνομα “appointment”, το event χαρακτηρίζεται με το μπλε χρώμα και θεωρείται σαν appointment. Εξίσου, αν επιλέξει το κουμπί με το όνομα

“other”, το γεγονός παίρνει το πράσινο χρώμα και χαρακτηρίζεται σαν other. Τέλος, αν ο χρήστης διαλέξει το κουμπί με το όνομα “anniversary”, το event αυτόματα θα πάρει για ώρα έναρξης την 00:01 και για ώρα λήξης την 23:59. Επίσης, θα χρωματιστεί με το κόκκινο χρώμα και θα χαρακτηριστεί σαν anniversary.

Ακολουθεί ο κώδικας για αυτά τα τρία κουμπιά.

```
appointment_button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (!start_time_input.isClickable() || !end_date_input.isClickable()){
            start_time_input.setClickable(true);
            end_time_input.setClickable(true);
        }
        event.setEventColor(Color.BLUE);
    }
});

//code that handles what happen if the button is pressed
anniversary_button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        //if the event is anniversary then the startDate should be 00:01
        //and the endDate should be 23:59. It's an allDay event
        calendar.setTimeInMillis(event.getStartDate());
        calendar.set(Calendar.HOUR_OF_DAY, 00);
        calendar.set(Calendar.MINUTE, 01);
        event.setStartDate(calendar.getTimeInMillis());
        start_time_text.setText("00:01");
        start_time_input.setClickable(false);
        end_time_input.setClickable(false);

        calendar.set(Calendar.HOUR_OF_DAY, 23);
        calendar.set(Calendar.MINUTE, 59);
        event.setEndDate(calendar.getTimeInMillis());
        end_time_text.setText("23:59");

        event.setEventColor(Color.RED);
    }
});

//code that handles what happen if the button is pressed
other_button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (!start_time_input.isClickable() || !end_date_input.isClickable()){
            start_time_input.setClickable(true);
            end_time_input.setClickable(true);
        }
        event.setEventColor(Color.GREEN);
    }
});
```

5.4.2.6 Cancel και Save buttons

Τα δύο αυτά κουμπιά καθορίζουν την τελική απόφαση του χρηστή. Πατώντας το κουμπί με το όνομα “Cancel” ακυρώνεται οποιαδήποτε διαδικασία έχει κάνει και επιστρέφει πίσω στο ημερολόγιο.

Ο κώδικας για το κουμπί Cancel έχει ως εξής.

```
cancel_button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        //return to the previous fragment
        FragmentManager fragmentManager =
getActivity().getSupportFragmentManager();
        if (fragmentManager.getBackStackEntryCount() > 0) {
            fragmentManager.popBackStack();
        }
    }
});
```

Όταν ο χρήστης πατήσει το κουμπί “Save” τότε γίνετε ένας έλεγχος για το αν έχουν συμπληρωθεί όλα τα πεδία. Σε περίπτωση που δεν έχουν συμπληρωθεί, εμφανίζεται στην οθόνη το μήνυμα “please complete all fields”. Αν όλα τα πεδία είναι σωστά συμπληρωμένα τότε το event αποθηκεύεται στη βάση, καλείται η συνάρτηση setEventNotification() και ο χρήστης επιστρέφεται στο ημερολόγιο. Η συνάρτηση setEventNotification() δημιουργεί μία ειδοποίηση για το event. Αν το event έχει χαρακτηριστεί σαν anniversary, τότε αυτό απλά αποθηκεύεται στη βάση, χωρίς να οριστεί κάποια ειδοποίηση.

Στην συνέχεια παρουσιάζεται ο κώδικας για το κουμπί Save.

```
save_button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        //creating of the event with the chosen values
        if (event_name_input.getText().toString().equals("") ||
                start_date_text.getText().toString().equals("Start Date") ||
                end_date_text.getText().toString().equals("End Date") ||
                start_time_text.getText().toString().equals("Start Time") ||
                end_time_text.getText().toString().equals("End Time") ||
                (!appointment_button.isChecked() && !
anniversary_button.isChecked() && !other_button.isChecked())){

            Toast.makeText(
                    getActivity().getApplicationContext(),
                    "please complete all fields",
                    Toast.LENGTH_LONG).show();
        }
```

```
else if (event.getStartDate()>event.getEndDate()){
            Toast.makeText(
                    getActivity().getApplicationContext(),
                    "endDate should be after startDate",
                    Toast.LENGTH_LONG).show();
        }
        else{
            event.setTitle(event_name_input.getText().toString());

            if (event.getEventColor() == Color.RED) {
                //if the event is an anniversary then just add it to the
database
                dbHandler.addEvent(event);
            }
            else {
                //if the event is not an anniversary then add it to the
database
                // and set the notification
                dbHandler.addEvent(event);
                setEventNotification(event);
            }

            String eventInfo = String.format("%tT - %tT: %s",
                    event.getStartDate(),
                    event.getEndDate(),
                    event.getTitle());

            Toast.makeText(
                    getActivity().getApplicationContext(),
                    eventInfo,
                    Toast.LENGTH_LONG).show();

            //when the event is created then return to the previous fragment
            FragmentManager fragmentManager =
getActivity().getSupportFragmentManager();
            if (fragmentManager.getBackStackEntryCount() > 0) {
                fragmentManager.popBackStack();
            }
        }
    }

});
```

5.4.3 setEventNotification() Method

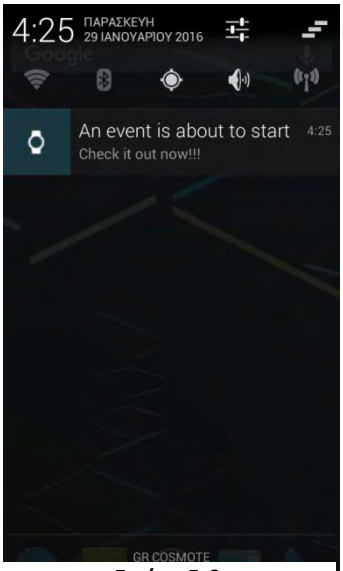
Η μέθοδος αυτή δημιουργεί μία ειδοποίηση για κάθε event που καταχωρείται στην βάση δεδομένων. Κατασκευάζεται ένα intent που εκτελεί την κλάση MyEventReceiver.java. Στη συνέχεια αρχικοποιείται ο AlarmManager και ορίζεται η ειδοποίηση στην ώρα έναρξης του event.

Ακολουθεί ο κώδικας της μεθόδου.

```
public void setEventNotification(Event event) {

    //When this intend is called, we want the MyEventReceiver.class to be
executed
    Intent myIntent2 = new Intent(getActivity().getApplicationContext(),
MyEventReceiver.class);
    myIntent2.setAction(Long.toString(System.currentTimeMillis()));
    int dummyUniqueInt = (int) System.currentTimeMillis(); //this variable
ensure as that the pendingIntent is unique
    PendingIntent pendingIntent2 =
PendingIntent.getBroadcast(getActivity().getApplicationContext(), 0, myIntent2,
0);

    //Initialize the notification manager, and set the notification
    AlarmManager alarmManager2 = (AlarmManager)
getActivity().getApplicationContext().getSystemService(Context.ALARM_SERVICE);
    alarmManager2.set(AlarmManager.RTC, event.getStartDate(), pendingIntent2);
}
```

5.5 MyEventReceiver

Αυτή η κλάση δημιουργεί την ειδοποίηση που καλείται από την μέθοδο setEventNotification(). Στην ουσία καθορίζει το τί θα εμφανιστεί στο notification bar, που βρίσκεται στο πάνω μέρος της οθόνης, όταν χτυπήσει η ειδοποίηση. Αρχικά, κατασκευάζεται μια λίστα με όλα τα events που υπάρχουν στη βάση. Αμέσως μετά, δημιουργείται ένα intent που εκτελεί την κλάση mainActivity.java και ένα PendingIntent που ενσωματώνει το intent αυτό. Η λειτουργία του PendingIntent είναι να εκτελεί το εξής intent όταν ο χρήστης πατήσει πάνω στην ειδοποίηση που θα του εμφανιστεί. Με λίγα λόγια, με το πάτημα της ειδοποίησης θα ανοίγει η εφαρμογή. Τέλος, ορίζεται το εικονίδιο της ειδοποίησης, ο τίτλος και το κείμενο της, καθώς και ο ήχος με την δόνηση που θα χτυπήσουν όταν αυτή ενεργοποιηθεί.



Εικόνα 5-6

Παρουσιάζεται ο κώδικας αυτής της κλάσης.

```
public class MyEventReceiver extends BroadcastReceiver {
    private NotificationManager mManager;

    @Override
    public void onReceive(Context context, Intent intent) {

        MyDBHandler dbHandler = new MyDBHandler(context, null, null, 1);
        List<Event> eventList = eventCreating(dbHandler);
        Calendar calendar = Calendar.getInstance();
        calendar.setTimeInMillis(System.currentTimeMillis());

        //Initialize the notification manager
        mManager = (NotificationManager)
context.getSystemService(Context.NOTIFICATION_SERVICE);

        //When this intend is called, we want the MainActivity.class to be
executed
        Intent intent1 = new Intent(context, MainActivity.class);
        intent1.addFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP |
Intent.FLAG_ACTIVITY_NEW_TASK);
        intent1.setAction(Long.toString(System.currentTimeMillis()));

        //The notification in the notification bar calls the intent1, which
mean that if we press the
        //notification shown in the notification bar, it will call the
MainActivity.class
        int dummyUniqueInt = (int) System.currentTimeMillis(); //this variable
ensure as that the pendingIntent is unique
        PendingIntent pendingNotificationIntent =
PendingIntent.getActivity(context, dummyUniqueInt, intent1,
PendingIntent.FLAG_UPDATE_CURRENT);

        //Creating the notification
        NotificationCompat.Builder builder = new
NotificationCompat.Builder(context);
        Notification notification =
builder.setContentIntent(pendingNotificationIntent)
                .setSmallIcon(R.drawable.ic_event)
                .setWhen(System.currentTimeMillis())
                .setAutoCancel(true)
                .setContentTitle("An event is about to start")
                .setContentText("Check it out now!!!")
                //.setContentText(getEventTitle(eventList))
//.setContentText(dbHandler.eventTitleRetrieving(System.currentTimeMillis()))
                .build();

        //The notification will have the default sound and vibrate
        notification.defaults |= Notification.DEFAULT_SOUND;
        notification.defaults |= Notification.DEFAULT_VIBRATE;
        mManager.notify(dummyUniqueInt, notification);
    }

    //Method that creates events that is stored in the database(TABLE_EVENTS)
    public List<Event> eventCreating(MyDBHandler dbHandler){
        final List<String> eventTitle = dbHandler.eventTitleRetrieving();
        final List<Long> eventStartTime = dbHandler.startTimeRetrieving();
        final List<Long> eventEndTime = dbHandler.endTimeRetrieving();
        List<Event> eventList = new ArrayList<Event>();
        for (int i = 0; i < eventTitle.size(); i++) {
            Event event = new Event(eventTitle.get(i), eventStartTime.get(i),
eventEndTime.get(i));
            eventList.add(event);
        }
        return eventList;
    }
}
```

Κεφάλαιο 6

6 Η λειτουργία RememberToDo

6.1 Περιγραφή της λειτουργίας

Επιλέγοντας την λειτουργία RememberToDo από το navigation drawer, ο χρήστης μεταβαίνει στο RememberFragment.java, όπου υπάρχει ένα πλαίσιο για εισαγωγή κειμένου και δύο κουμπιά. Το ένα κουμπί εμφανίζεται με το σύμβολο της πρόσθεσης και το άλλο με το σύμβολο της αφαίρεσης.

Όταν ο χρήστης πληκτρολογήσει ένα task στο πλαίσιο κειμένου και πατήσει το κουμπί με το σύμβολο της πρόσθεσης, αυτό το task αποθηκεύεται στην βάση δεδομένων της εφαρμογής και εμφανίζεται σε μορφή λίστας ακριβώς κάτω από το πλαίσιο εισαγωγής κειμένου με ένα checkbox δίπλα του.

Εικόνα 6-1

Αν θεωρηθεί από το χρήστη πως κάποια από τα task που έχουν προστεθεί στη λίστα, έχουν ολοκληρωθεί μπορεί να τα τσεκάρει με ένα απλό άγγιγμα.

Για την διαγραφή των tasks από την λίστα, υπάρχουν δύο επιλογές. Είτε ο χρήστης να τσεκάρει τα tasks προς διαγραφή και να πατήσει το κουμπί με το σύμβολο της αφαίρεσης, είτε να αγγίξει κάποιο task για αρκετή ώρα. Όταν ένα task διαγραφεί από την λίστα, τότε διαγράφεται αυτόματα και από την βάση.

6.2 RememberFragment

Σε αυτή την κλάση υλοποιούνται όλα όσα αναφέρθηκαν παραπάνω. Όπως και σε όλα τα άλλα fragments, όλες οι λειτουργίες ορίζονται μέσα στη μέθοδο onViewCreated().

6.2.1 onCreatedView() Method

Εδώ γίνεται η σύνδεση των γραφικών που περιγράφονται στο αρχείο fragment_remember.xml με την κλάση.

Παρουσιάζεται ο κώδικας της μεθόδου,

```
public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
    final View rootView = inflater.inflate(R.layout.fragment_remember,
container, false);
    return rootView;
}
```

και ο κώδικας του .xml αρχείου.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    tools:context=".RememberFragment"
    android:id="@+id/fragment_remember"
    android:weightSum="1">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal" >

        <EditText
            android:id="@+id/task_input"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="0.67"
            android:ems="10"
            android:inputType="text"
            android:hint="@string/input_task">
            <requestFocus />
        </EditText>

        <Button
            android:id="@+id/submit_button"
            android:layout_width="47dp"
            android:layout_height="wrap_content"
            android:text="@string/add_button" />

        <Button
            android:layout_width="47dp"
            android:layout_height="wrap_content"
            android:text="@string/delete_button"
            android:id="@+id/delete_button" />
    </LinearLayout>
    <ListView
        android:id="@+id/task_list"
        android:layout_width="match_parent"
        android:layout_height="409dp"
        android:layout_weight="0.23"
        />
</LinearLayout>
```

6.2.2 onViewCreated() Method

Στην αρχή αυτή της μεθόδου γίνεται η σύνδεση των στοιχείων του .xml αρχείου με τις μεταβλητές της κλάσης. Στην συνέχεια δημιουργείται μια λίστα που θα κρατάει τα tasks που είναι αποθηκευμένα στη βάση. Τα στοιχεία αυτής της λίστας (tasks) έχουν την μορφή όπως την ορίζει το listview_task_row.xml αρχείο.

Ακολουθεί ο κώδικας όσων αναφέρθηκαν,

```
task_input = (EditText) rootView.findViewById(R.id.task_input);
task_list = (ListView) rootView.findViewById(R.id.task_list);
submit_button = (Button) rootView.findViewById(R.id.submit_button);
delete_button = (Button) rootView.findViewById(R.id.delete_button);

final List<ObjectListTask> mTasks = new ArrayList<ObjectListTask>();
final ListAdapter adapter = new
ListTaskCustomAdapter(getActivity().getApplicationContext(),R.layout.listview_t
ask_row, mTasks);
task_list.setAdapter(adapter);
```

και ο κώδικας του .xml αρχείου.

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="?android:attr/activatedBackgroundIndicator"
    android:minHeight="?android:attr/listPreferredItemHeightSmall"
    android:padding="10dp"
    android:descendantFocusability="blocksDescendants">

    <TextView
        android:id="@+id/textViewTaskName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerVertical="true"
        android:layout_alignParentLeft="true"
        android:layout_toLeftOf="@+id/checkbox"
        android:paddingRight="10dp"
        android:text="Task name here."
        android:textColor="#000000"
        />

    <CheckBox
        android:id="@+id/checkbox"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:paddingRight="10dp"
        android:buttonTint="#000000"
        android:clickable="false" />
</RelativeLayout>
```

Έπειτα, διαβάζονται τα tasks που είναι αποθηκευμένα στη βάση δεδομένων και φορτώνονται στην λίστα που ορίστηκε παραπάνω.

```
dbHandler = new MyDBHandler(getActivity().getApplicationContext(), null, null,
1);
final List<String> dbString = dbHandler.databaseToString();     //get the
tasknames from the database
final List<Boolean> dbChecked = dbHandler.databaseIsChecked();      //get the
value of each checkbox of the task
//->final ObjectListTask[] intasks = new ObjectListTask[dbString.size()];
for (int i=0; i<dbString.size(); i++) {
    //->intasks[i] = new ObjectListTask(dbChecked.get(i),
dbString.get(i).toString());
    ObjectListTask intasks = new ObjectListTask(dbChecked.get(i),
dbString.get(i).toString());
    //->mTasks.add(intasks[i]);
    mTasks.add(intasks);    //create the input task (checkbox, name) and added
to the ListView
}

task_list.setAdapter(adapter);
task_input.setText("");
```

Ο κώδικας παρακάτω αποτελεί την διασύνδεση με τον χρήστη. Όταν ο χρήστης πατήσει πάνω σε ένα στοιχείο της λίστας τότε αυτό από checked γίνεται unchecked ή το αντίθετο. Όταν συμβεί αυτό η αλλαγή γίνεται αυτόματα και στη βάση δεδομένων.

```
task_list.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position,
long id) {
        mTasks.get(position).setChecked(!(mTasks.get(position).isChecked()));
//gives to the checkbox the opposite value
        dbHandler.updateChecked(mTasks.get(position));      //set the new value
of the checkbox to the database
        task_list.setAdapter(adapter);
        //task_input.setText("");
    }
});
```

Αν ο χρήστης κρατήσει πατημένο παρατεταμένα το στοιχείο της λίστας, τότε αυτό διαγράφεται από την λίστα αλλά και από την βάση δεδομένων.

```
task_list.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener()
{
    @Override
    public boolean onItemLongClick(AdapterView<?> parent, View view, int
position, long id) {
        dbHandler.deleteTask(mTasks.get(position).getName());   //delete the
task from the database
        mTasks.remove(mTasks.get(position));    //delete the task from the
ListView
        task_list.setAdapter(adapter);
        //task_input.setText("");

        return true;
    }
});
```

Πατώντας στο κουμπί για την πρόσθεση, ό, τι είναι γραμμένο στο πλαίσιο κειμένου, αποθηκεύεται στη βάση δεδομένων και εμφανίζεται στη λίστα.

```
submit_button.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View v) {

        // Add a new task to the database.
        ObjectListTask intask = new ObjectListTask(Boolean.FALSE,
task_input.getText().toString());
        dbHandler.addTask(intask);  //set the intask in the database
        mTasks.add(intask);         //set the intask in the ListView
        task_list.setAdapter(adapter);
        task_input.setText("");

    }
});
```

Τέλος, όταν ο χρήστης πατήσει το κουμπί διαγραφής, τότε όποιο στοιχείο της λίστας είναι checked διαγράφεται από την λίστα και από την βάση δεδομένων ταυτόχρονα.

```
delete_button.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {
            // Delete the checked tasks from the database.

                dbHandler.deleteTask();     //delete the checked tasks
                onViewCreated(rootView, savedInstanceState);    //call again
the onViewCreated method,so the mTasks get initialized again
            }
        });
```

Κεφάλαιο 7

7 Η λειτουργία Settings

Εικόνα 6-1

7.1 Περιγραφή της λειτουργίας

Η λειτουργία Settings προσδιορίζεται από την κλάση SettingsFragment.java. Στο εσωτερικό αυτής της κλάσης υλοποιούνται δύο υπολειτουργίες ρυθμίσεων που είναι παρόμοιες μεταξύ τους.

Η πρώτη υπολειτουργία ονομάζεται “Schedule Notification Settings”, και ρυθμίζει την ώρα που θέλει ο χρήστης να ειδοποιείται για το ημερήσιο πρόγραμμα του. Η δεύτερη υπολειτουργία φέρει το όνομα “Anniversary Notification Settings” και κατ’ αντιστοιχία, ρυθμίζει την ώρα ειδοποίησης για τα ημερήσια events που έχουν χαρακτηριστεί ως anniversary. Και οι δύο αυτές ρυθμίσεις είναι προαιρετικές.

7.2 SettingsFragment

7.2.1 onCreateView() Method

Το αρχείο fragment_settings.xml περιέχει τα γραφικά που χρησιμοποιεί αυτή η μέθοδος, που δεν είναι τίποτα άλλο από μια λίστα. Σε αυτή τη μέθοδο γίνετε η σύνδεση τους με την κλάση.

```
public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
    View rootView = inflater.inflate(R.layout.fragment_settings, container,
false);
    return rootView;
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context=".SettingsFragment"
    android:id="@+id/settings">

    <ListView
    android:id="@+id/settings_list"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
</FrameLayout>
```

7.2.2 onViewCreated() Method

Όπως και σε κάθε άλλο fragment, στο εσωτερικό αυτής της κλάσης υλοποιούνται οι λειτουργίες της. Ορίζεται ένας πίνακας από String που κρατάει τα ονόματα των γραμμών της λίστας. Ο πίνακας αυτός συνδέεται με την λίστα μέσω ενός ArrayAdapter και του listview_setting_row.xml αρχείου όπου δηλώνεται η δομή των γραμμών της λίστας.

```
settingsListView = (ListView) rootView.findViewById(R.id.settings_list);
dbHandler = new MyDBHandler(getActivity().getApplicationContext(), null, null,
1);
String[] settings = new String[] {"Schedule Notification Settings",
"Anniversary Notification Settings"};
ArrayList<String> settingsList = new ArrayList<String>();
settingsList.addAll(Arrays.asList(settings));

//Creating the listView in the settings fragment
listAdapter = new ArrayAdapter<String>(getActivity().getApplicationContext(),
R.layout.listview_setting_row, R.id.setting_row, settingsList);
settingsListView.setAdapter(listAdapter);
```

Παρουσιάζεται ο κώδικας του listview_setting_row.xml

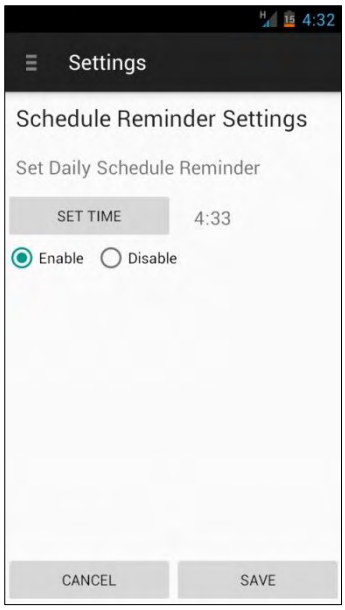
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <TextView
        android:id="@+id/setting_row"
        android:layout_width="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_height="wrap_content"
        android:textColor="#000000"
        android:padding="10dp"
        android:textAppearance="?android:attr/textAppearanceMedium" />
    </RelativeLayout>
```

Στη συνέχεια ακολουθεί ο κώδικας που ορίζει τι θα συμβεί όταν ο χρήστης πατήσει πάνω σε μία γραμμή της λίστας.

Το πάτημα του πρώτου στοιχείου της λίστας δημιουργεί το ScheduleSettingFragment που όπως έχει προαναφερθεί, οδηγεί στις ρυθμίσεις για την ειδοποίηση του ημερήσιου προγράμματος. Αν ο χρήστης επιλέξει το δεύτερο στοιχείο της λίστας, θα δημιουργήσει το AnniversarySettingFragment και θα μεταβεί στις ρυθμίσεις για την ειδοποίηση των ημερήσιων anniversary γεγονότων.

```
settingsListView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position,
long id) {

        Fragment fragment = null;

        switch (position) {
            case 0:
                fragment = new ScheduleSettingsFragment();
                break;
            case 1:
                fragment = new AnniversarySettingsFragment();
                break;
            case 2:

            default:
                break;
        }

        if (fragment != null) {
            FragmentManager fragmentManager =
getActivity().getSupportFragmentManager();
            fragmentManager.beginTransaction().replace(R.id.content_frame,
fragment).addToBackStack(null).commit();
        }
        else{
            Log.e("SettingsFragment", "Error in creating fragment");
        }
    }
});
```

Εικόνα 7-2



7.3 ScheduleSettingFragment

Μέσα από αυτή την κλάση ο χρήστης έχει την δυνατότητα να ενεργοποιήσει/απενεργοποιήσει την λειτουργία ειδοποίησης για τον έλεγχο του καθημερινού προγράμματος του καθώς και να επιλέξει την ώρα αυτής της ειδοποίησης. Η λειτουργία εξ' ορισμού είναι απενεργοποιημένη και η ώρα χτυπήματος είναι στις 8:00.

7.3.1 onCreateView() Method

Σε αυτή την κλάση γίνεται η σύνδεση των γραφικών του αρχείου fragment_schedule_settings.xml με την κλάση.

```
public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
    View rootView = inflater.inflate(R.layout.fragment_schedule_settings,
container, false);
    return rootView;
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:id="@+id/fragment_schedule_settings"
    tools:context=".ScheduleSettingsFragment"
    android:weightSum="10">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/schedule_settings_title"
        android:id="@+id/schedule_settings_title"
        android:padding="10dp"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:allowUndo="false" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:text="@string/schedule_settings_instruction"
        android:id="@+id/schedule_settings_instruction"
        android:padding="10dp"
        android:layout_below="@+id/schedule_settings_title"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:text="@string/sch_reminder_time_input"
        android:id="@+id/sch_reminder_time_input"
        android:padding="10dp"
        android:layout_below="@+id/schedule_settings_instruction"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />
```

```
<TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:padding="10dp"
        android:text="@string/sch_reminder_time_text"
        android:id="@+id/sch_reminder_time_text"
        android:layout_alignBottom="@+id/sch_reminder_time_input"
        android:layout_toRightOf="@+id/sch_reminder_time_input"
        android:layout_toEndOf="@+id/sch_reminder_time_input" />

    <RadioGroup
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/sch_reminder_time_text"
        android:id="@+id/sch_reminder_state"
        android:orientation="horizontal">

        <RadioButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/enable_button"
            android:text="@string/enable_button"
            android:checked="false"
            android:layout_marginRight="10dp"
            android:singleLine="true" />

        <RadioButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/disable_button"
            android:text="@string/disable_button"
            android:layout_marginRight="10dp"
            android:checked="false" />
    </RadioGroup>
    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:weightSum="1"
        android:layout_below="@+id/sch_reminder_state">

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/cancel_button"
            android:id="@+id/cancel_button"
            android:layout_weight="0.50"
            android:layout_gravity="bottom" />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/save_button"
            android:id="@+id/save_button"
            android:layout_weight="0.50"
            android:layout_gravity="bottom" />
    </LinearLayout>
</RelativeLayout>
```

7.3.2 onViewCreated() Method

Τα στοιχεία του .xml αρχείου εκχωρούνται σε μεταβλητές ώστε να μπορούν να διαχειριστούν. Διαβάζονται τα αποθηκευμένα δεδομένα της βάσης δεδομένων (στην ενότητα 4.3 έχει αναφερθεί πως οι πίνακες της λειτουργίας Settings έχουν μόνο μία εγγραφή και έχει γίνει η αρχικοποίηση τους) και δημιουργείται ένα αντικείμενο τύπου ScheduleSettings όπως ορίζεται από τον constructor της κλάσης ScheduleSettings.java.

Παρατίθεται ο κώδικας των όσων αναφέρθηκαν,

```
setTimeButton = (Button) rootView.findViewById(R.id.sch_reminder_time_input);
setTimeText = (TextView) rootView.findViewById(R.id.sch_reminder_time_text);
enableButton = (RadioButton) rootView.findViewById(R.id.enable_button);
disableButton = (RadioButton) rootView.findViewById(R.id.disable_button);
cancelButton = (Button) rootView.findViewById(R.id.cancel_button);
saveButton = (Button) rootView.findViewById(R.id.save_button);
final Calendar calendar = Calendar.getInstance();
final MyDBHandler dbHandler = new
MyDBHandler(getActivity().getApplicationContext(), null, null, 1);
final long time = dbHandler.timeScheduleRetrieving();
final int state = dbHandler.stateScheduleRetrieving();
final ScheduleSettings settings = new ScheduleSettings(time, state);

//Take the stored time of the database(TABLE_ANNIVERSARY_SETTINGS) and set it
to the TextView
String timeView;
calendar.setTimeInMillis(settings.getTime());
if (calendar.get(Calendar.MINUTE)<10){
    timeView = (calendar.get(Calendar.HOUR_OF_DAY) + ":0" +
calendar.get(Calendar.MINUTE));
}
else{
    timeView = (calendar.get(Calendar.HOUR_OF_DAY) + ":" +
calendar.get(Calendar.MINUTE));
}
setTimeText.setText(timeView);

//Take the stored state of the database(TABLE_ANNIVERSARY_SETTINGS) and set it
to the RadioButton
if (settings.getState() == 1){
    enableButton.setChecked(true);
    disableButton.setChecked(false);
}
else if (settings.getState() == 2) {
    enableButton.setChecked(false);
    disableButton.setChecked(true);
}
else {
    Toast.makeText(
            getActivity().getApplicationContext(),
            "Error getting the settings state!",
            Toast.LENGTH_LONG).show();
}
```

και ο κώδικας της κλάσης ScheduleSettings.java

```
public class ScheduleSettings {

    long time;
    int state;

    public ScheduleSettings() {

    }

    //Creating an object of the class
    public ScheduleSettings(long time, int state) {
        this.time = time;
        this.state = state;
    }

    public void setTime(long time) {
        this.time = time;
    }

    public void setState(int state) {
        this.state = state;
    }

    public int getState() {
        return state;
    }

    public long getTime() {
        return time;
    }
}
```

7.3.2.1 Set time button

Όταν ο χρήστης πατήσει το κουμπί “Set Time”, καλείται η μέθοδος SelectTimeFragment() της αντίστοιχης κλάσης και εμφανίζεται ένα dialog παράθυρο για την επιλογή ώρας. Με αυτό το κουμπί ο χρήστης επιλέγει τι ώρα θέλει να ειδοποιείται για το πρόγραμμα του.

```
setTimeButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        //showDatePicker();
        //Creating the Select Time dialog
        SelectTimeFragment date = new SelectTimeFragment();

        //Initialize the dialog with the stored values of time
        Bundle args = new Bundle();
        args.putInt("hour", calendar.get(Calendar.HOUR_OF_DAY));
        args.putInt("minutes", calendar.get(Calendar.MINUTE));
        date.setArguments(args);
        date.setCallBack(ondate);
        date.show(getChildFragmentManager(), "Time Picker");
    }

    TimePickerDialog.OnTimeSetListener ondate = new
TimePickerDialog.OnTimeSetListener() {
        @Override
        public void onTimeSet(TimePicker view, int hourOfDay, int minute) {

            //Set to the TextView the selected time
            String time;
            if (minute < 10) {
                time = (hourOfDay + ":0" + minute);
            } else {
                time = (hourOfDay + ":" + minute);
            }
            setTimeText.setText(time);

            //Set to the settings object the selected time
            calendar.set(Calendar.HOUR_OF_DAY, hourOfDay);
            calendar.set(Calendar.MINUTE, minute);
            settings.setTime(calendar.getTimeInMillis());

        }
    };
});
```

7.3.2.2 Enable/Disable buttons

Αυτά τα δύο κουμπιά ορίζουν την λειτουργία της ειδοποίησης. Όταν τσεκαριστεί το κουμπί enable η ειδοποίηση θα λειτουργεί κανονικά και το κουμπί θα disable δεν θα είναι επιλεγμένο. Το αντίθετο συμβαίνει όταν επιλεγεί το κουμπί disable.

```
enableButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        settings.setState(1);
    }
});
```

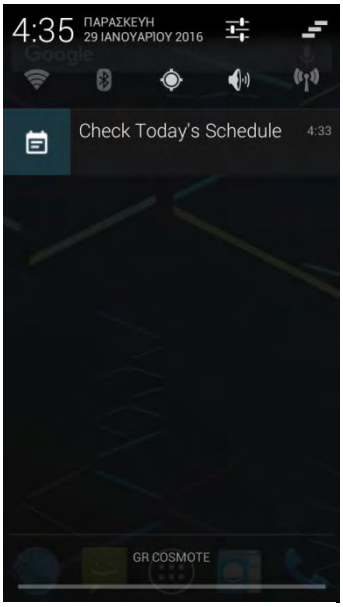
```
disableButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        settings.setState(2);
    }
});
```

7.3.2.3 Cancel και Save buttons

Πατώντας το κουμπί “Cancel” τότε ακυρώνεται όποια διαδικασία έχει ακολουθηθεί και ο χρήστης επιστρέφεται στο fragment της λειτουργίας Settings.

Ακολουθεί ο κώδικας για το κουμπί Cancel.

```
cancelButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        //return to the previous fragment
        FragmentManager fragmentManager =
getActivity().getSupportFragmentManager();
        if (fragmentManager.getBackStackEntryCount() > 0) {
            fragmentManager.popBackStack();
        }
    }
});
```



Εικόνα 7-3

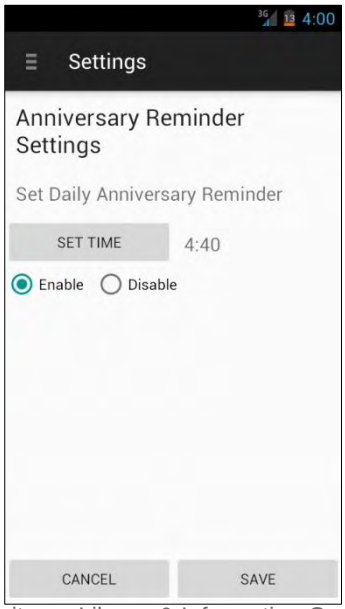
Αν ο χρήστης επιλέξει το κουμπί “Save”, οι αλλαγές που έχουν γίνει, αποθηκεύονται στη βάση δεδομένων και ορίζεται η ειδοποίηση για την συγκεκριμένη ώρα.

Παρατίθεται ο κώδικας για το κουμπί “Save”,

```
saveButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        //Add to the database the new values
        dbHandler.setScheduleTime(settings.getTime());
        dbHandler.setScheduleState(settings.getState());

        //Creating the schedule notification
        //When this intend is called, we want the MyReceiver.class to be
executed
        Intent myIntent = new Intent(getActivity().getApplicationContext(),
MyReceiver.class);
        PendingIntent pendingIntent =
PendingIntent.getBroadcast(getActivity().getApplicationContext(), 0, myIntent,
0);

        AlarmManager alarmManager = (AlarmManager)
getActivity().getApplicationContext().getSystemService(Context.ALARM_SERVICE);

        if (settings.getState() == 1) {
            //set the schedule notification
            alarmManager.setRepeating(AlarmManager.RTC,
calendar.getTimeInMillis(), AlarmManager.INTERVAL_DAY, pendingIntent);
        }
        else if (settings.getState() == 2) {
            //cancel the schedule notification
            alarmManager.cancel(pendingIntent);
        }
        else {
            Toast.makeText(
                    getActivity().getApplicationContext(),
                    "Error during retrieving the notification state",
                    Toast.LENGTH_LONG).show();
        }

        //return to the previous fragment
        FragmentManager fragmentManager =
getActivity().getSupportFragmentManager();
        if (fragmentManager.getBackStackEntryCount() > 0) {
            fragmentManager.popBackStack();
        }
    }
});
```

και ο κώδικας της κλάσης MyReceiver.java που ορίζει τα χαρακτηριστικά της ειδοποίησης.

```
public class MyReceiver extends BroadcastReceiver {
    private NotificationManager mManager;

    @Override
    public void onReceive(Context context, Intent intent) {

        //Initialize the notification manager
        mManager = (NotificationManager)
context.getSystemService(Context.NOTIFICATION_SERVICE);

        //When this intend is called, we want the MainActivity.class to be
executed
        Intent intent1 = new Intent(context,MainActivity.class);
        intent1.addFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP|
Intent.FLAG_ACTIVITY_CLEAR_TOP);

        //The notification in the notification bar calls the intent1, which
mean that if we press the
        //notification shown in the notification bar, it will call the
MainActivity.class
        PendingIntent pendingNotificationIntent =
PendingIntent.getActivity(context,0,
intent1,PendingIntent.FLAG_UPDATE_CURRENT);

        //Creating the notification
        NotificationCompat.Builder builder = new
NotificationCompat.Builder(context);
        Notification notification =
builder.setContentIntent(pendingNotificationIntent)
                .setSmallIcon(R.drawable.ic_schedule)
                .setWhen(System.currentTimeMillis())
                .setAutoCancel(true)
                .setContentTitle("Check Today's Schedule")
                .build();

        //The notification will have the default sound and vibrate
        notification.defaults |= Notification.DEFAULT_SOUND;
        notification.defaults |= Notification.DEFAULT_VIBRATE;
        mManager.notify(0, notification);

    }
}
```



Εικόνα 7-4

7.4 AnniversarySettingsFragment

Η λειτουργία αυτής της υπολειτουργίας είναι παρόμοια με αυτή που περιγράφτηκε στην προηγούμενη παράγραφο. Ο χρήστης ρυθμίζει την ειδοποίηση των καθημερινών anniversary γεγονότων.

7.4.1 onCreateView() Method

Αυτή η μέθοδος δεν παρουσιάζει καμία διαφορά.

```
public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
    View rootView = inflater.inflate(R.layout.fragment_anniversary_settings,
container, false);
    return rootView;
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:id="@+id/fragment_anniversary_settings"
    tools:context=".AnniversarySettingsFragment"
    android:weightSum="10">


    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/anniversary_settings_title"
        android:id="@+id/anniversary_settings_title"
        android:padding="10dp"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:allowUndo="false" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:text="@string/anniversary_settings_instruction"
        android:id="@+id/anniversary_settings_instruction"
        android:padding="10dp"
        android:layout_below="@+id/anniversary_settings_title"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:text="@string/ann_reminder_time_input"
        android:id="@+id/ann_reminder_time_input"
        android:padding="10dp"
        android:layout_below="@+id/anniversary_settings_instruction"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />
```

```
<TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:padding="10dp"
        android:text="@string/ann_reminder_time_text"
        android:id="@+id/ann_reminder_time_text"
        android:layout_alignBottom="@+id/ann_reminder_time_input"
        android:layout_toRightOf="@+id/ann_reminder_time_input"
        android:layout_toEndOf="@+id/ann_reminder_time_input" />

    <RadioGroup
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/ann_reminder_time_text"
        android:id="@+id/ann_reminder_state"
        android:orientation="horizontal">

        <RadioButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/ann_enable_button"
            android:text="@string/ann_enable_button"
            android:checked="false"
            android:layout_marginRight="10dp"
            android:singleLine="true" />

        <RadioButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/ann_disable_button"
            android:text="@string/ann_disable_button"
            android:layout_marginRight="10dp"
            android:checked="false" />

    </RadioGroup>
    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:weightSum="1"
        android:layout_below="@+id/ann_reminder_state">

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/cancel_button"
            android:id="@+id/cancel_button"
            android:layout_weight="0.50"
            android:layout_gravity="bottom" />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/save_button"
            android:id="@+id/save_button"
            android:layout_weight="0.50"
            android:layout_gravity="bottom" />
    </LinearLayout>
</RelativeLayout>
```

7.4.2 onViewCreated() Method

```
setTimeButton = (Button) rootView.findViewById(R.id.ann_reminder_time_input);
setTimeText = (TextView) rootView.findViewById(R.id.ann_reminder_time_text);
enableButton = (RadioButton) rootView.findViewById(R.id.ann_enable_button);
disableButton = (RadioButton) rootView.findViewById(R.id.ann_disable_button);
cancelButton = (Button) rootView.findViewById(R.id.cancel_button);
saveButton = (Button) rootView.findViewById(R.id.save_button);
final Calendar calendar = Calendar.getInstance();
final MyDBHandler dbHandler = new
MyDBHandler(getActivity().getApplicationContext(), null, null, 1);
final long time = dbHandler.timeAnniversaryRetrieving();
final int state = dbHandler.stateAnniversaryRetrieving();
final AnniversarySettings settings = new AnniversarySettings(time, state);

//Take the stored time of the database(TABLE_ANNIVERSARY_SETTINGS) and set it
to the TextView
String timeView;
calendar.setTimeInMillis(settings.getTime());
if (calendar.get(Calendar.MINUTE)<10){
    timeView = (calendar.get(Calendar.HOUR_OF_DAY) + ":0" +
calendar.get(Calendar.MINUTE));
}
else{
    timeView = (calendar.get(Calendar.HOUR_OF_DAY) + ":" +
calendar.get(Calendar.MINUTE));
}
setTimeText.setText(timeView);

//Take the stored state of the database(TABLE_ANNIVERSARY_SETTINGS) and set it
to the RadioButton
if (settings.getState() == 1){
    enableButton.setChecked(true);
    disableButton.setChecked(false);
}
else if (settings.getState() == 2) {
    enableButton.setChecked(false);
    disableButton.setChecked(true);
}
else {
    Toast.makeText(
            getActivity().getApplicationContext(),
            "Error getting the settings state!",
            Toast.LENGTH_LONG).show();
}
```

Στη συνέχεια είναι ο κώδικας της κλάσης AnniversarySettings.java

```
public class AnniversarySettings {

    long time;
    int state;

    public AnniversarySettings() {
    }

    //Creating an object of the class
    public AnniversarySettings(long time, int state) {
        this.time = time;
        this.state = state;
    }

    public void setTime(long time) {
        this.time = time;
    }

    public void setState(int state) {
        this.state = state;
    }

    public int getState() {
        return state;
    }

    public long getTime() {
        return time;
    }
}
```

7.4.2.1 Set time button

Όταν ο χρήστης πατήσει το κουμπί “Set Time” εκτελείται ο παρακάτω κώδικας.

```
setTimeButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        //showDatePicker();
        //Creating the Select Time dialog
        SelectTimeFragment date = new SelectTimeFragment();

        //Initialize the dialog with the stored values of time
        Bundle args = new Bundle();
        args.putInt("hour", calendar.get(Calendar.HOUR_OF_DAY));
        args.putInt("minutes", calendar.get(Calendar.MINUTE));
        date.setArguments(args);
        date.setCallBack(ondate);
        date.show(getChildFragmentManager(), "Time Picker");
    }

    TimePickerDialog.OnTimeSetListener ondate = new
TimePickerDialog.OnTimeSetListener() {
        @Override
        public void onTimeSet(TimePicker view, int hourOfDay, int minute) {

            //Set to the TextView the selected time
            String time;
            if (minute < 10) {
                time = (hourOfDay + ":0" + minute);
            } else {
                time = (hourOfDay + ":" + minute);
            }
            setTimeText.setText(time);

            //Set to the settings object the selected time
            calendar.set(Calendar.HOUR_OF_DAY, hourOfDay);
            calendar.set(Calendar.MINUTE, minute);
            settings.setTime(calendar.getTimeInMillis());
        }
    };
});
```

7.4.2.2 Enable/Disable buttons

Ο κώδικας του κουμπιού “enable” και “disable” αντίστοιχα.

```
enableButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        settings.setState(1);
    }
});

disableButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        settings.setState(2);
    }
});
```

7.4.2.3 Cancel και Save buttons

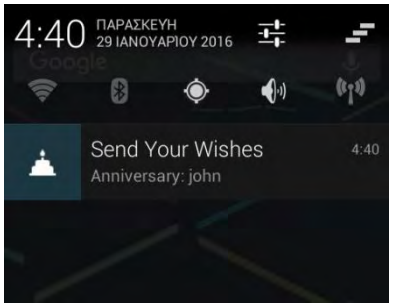
Παρουσιάζεται ο κώδικας για τα κουμπιά “Cancel” και “Save”.

```
cancelButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        //return to the previous fragment
        FragmentManager fragmentManager =
getActivity().getSupportFragmentManager();
        if (fragmentManager.getBackStackEntryCount() > 0) {
            fragmentManager.popBackStack();
        }
    }
});

saveButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        //Add to the database the new values
        dbHandler.setAnniversaryTime(settings.getTime());
        dbHandler.setAnniversaryState(settings.getState());

        //Creating the anniversary notification
        //When this intend is called, we want the MyAnniversaryReceiver.class
to be executed
        Intent myIntent = new Intent(getActivity().getApplicationContext(),
MyAnniversaryReceiver.class);
        PendingIntent pendingIntent =
PendingIntent.getBroadcast(getActivity().getApplicationContext(), 0, myIntent,
0);

        AlarmManager alarmManager = (AlarmManager)
getActivity().getApplicationContext().getSystemService(Context.ALARM_SERVICE);

        if (settings.getState() == 1) {
            //set the anniversary notification
            alarmManager.setRepeating(AlarmManager.RTC,
calendar.getTimeInMillis(), AlarmManager.INTERVAL_DAY, pendingIntent);
        }
        else if (settings.getState() == 2) {
            //cancel the anniversary notification
            alarmManager.cancel(pendingIntent);
        }
        else {
            Toast.makeText(
                    getActivity().getApplicationContext(),
                    "Error during retrieving the notification state",
                    Toast.LENGTH_LONG).show();
        }

        //return to the previous fragment
        FragmentManager fragmentManager =
getActivity().getSupportFragmentManager();
        if (fragmentManager.getBackStackEntryCount() > 0) {
            fragmentManager.popBackStack();
        }
    }
});
```

Και τέλος ο κώδικας της κλάσης MyAnniversaryReceiver.java.

```
public class MyAnniversaryReceiver extends BroadcastReceiver {
    private NotificationManager mManager;

    @Override
    public void onReceive(Context context, Intent intent) {

        MyDBHandler dbHandler = new MyDBHandler(context, null, null, 1);
        List<Event> eventList = eventCreating(dbHandler);

        //Initialize the notification manager
        mManager = (NotificationManager)
context.getSystemService(Context.NOTIFICATION_SERVICE);

        //When this intend is called, we want the MainActivity.class to be
executed
        Intent intent1 = new Intent(context, MainActivity.class);
        intent1.addFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP |
Intent.FLAG_ACTIVITY_CLEAR_TOP);

        //The notification in the notification bar calls the intent1, which
mean that if we press the
        //notification shown in the notification bar, it will call the
MainActivity.class
        PendingIntent pendingNotificationIntent =
PendingIntent.getActivity(context, 0, intent1,
PendingIntent.FLAG_UPDATE_CURRENT);

        //Creating the notification
        NotificationCompat.Builder builder = new
NotificationCompat.Builder(context);
        Notification notification =
builder.setContentIntent(pendingNotificationIntent)
                .setSmallIcon(R.drawable.ic_anniversary)
                .setWhen(System.currentTimeMillis())
                .setAutoCancel(true)
                .setContentTitle("Send Your Wishes")
                .setContentText("Anniversary: " + getEventTitle(eventList))
                .build();


        //The notification will have the default sound and vibrate
        notification.defaults |= Notification.DEFAULT_SOUND;
        notification.defaults |= Notification.DEFAULT_VIBRATE;

        mManager.notify(1, notification);

    }
```

```
public List<Event> eventCreating(MyDBHandler dbHandler){
        Calendar eventStartDate = Calendar.getInstance();
        Calendar calendar = Calendar.getInstance();
        calendar.setTimeInMillis(System.currentTimeMillis());
        long eventStartMonth = calendar.get(Calendar.MONTH);
        long eventStartDay = calendar.get(Calendar.DAY_OF_MONTH);

        List<String> eventTitle = dbHandler.eventTitleRetrieving();
        List<Long> eventStartTime = dbHandler.startTimeRetrieving();
        List<Long> eventEndTime = dbHandler.endTimeRetrieving();
        List<Integer> eventColor = dbHandler.ColorRetrieving();
        List<Event> eventList = new ArrayList<Event>();

        for (int i = 0; i < eventTitle.size(); i++) {
            Event event = new Event(eventTitle.get(i), eventStartTime.get(i),
eventEndTime.get(i));
            event.setEventColor(eventColor.get(i));

            eventStartDate.setTimeInMillis(event.getStartDate());
            if (event.getEventColor() == Color.RED &&
                    eventStartDate.get(Calendar.MONTH) == eventStartMonth &&
                    eventStartDate.get(Calendar.DAY_OF_MONTH) == eventStartDay)
{

                eventList.add(event);
            }
        }

        return eventList;
    }

    //Method that returns the titles of all events in a string
    public String getEventTitle(List<Event> eventList){

        String title = "";
        for (int i=0; i<eventList.size(); i++){
            title += eventList.get(i).getTitle();
            title += "\n";
        }
        return title;
    }

}
```



Εικόνα 7-5

Στην διπλανή εικόνα φαίνεται πως έχει οριστεί η ειδοποίηση από την κλάση MyAnniversaryReceiver.java

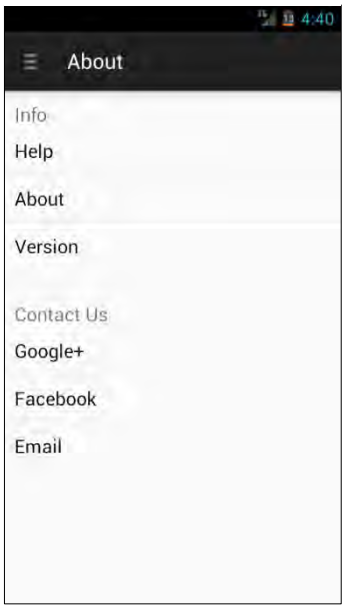
Κεφάλαιο 8

8 Η λειτουργία About

8.1 Περιγραφή της λειτουργίας

Αυτή η λειτουργία προσδιορίζεται από την κλάση AboutFragment.java. Σκοπός της είναι να δώσει στο χρήστη πληροφορίες για την εφαρμογή. Το γραφικό περιβάλλον που παρέχει αυτή η κλάση αποτελείται από δύο λίστες.

Εικόνα 8-1



Η πρώτη λίστα περιέχει τρία στοιχεία. Το πρώτο στοιχείο με το όνομα “Help” παρέχει πληροφορίες για τον τρόπο λειτουργίας της εφαρμογής. Το δεύτερο στοιχείο ονομάζεται “About” και αναφέρεται στον προγραμματιστή. Το τελευταίο στοιχείο της πρώτης λίστας φέρει το όνομα “Version” και έχει πληροφορίες για την έκδοση της εφαρμογής.

Η δεύτερη λίστα, όπως και η πρώτη, έχει και αυτή τρία στοιχεία. Τα δύο πρώτα με όνομα “Google+” και “Facebook” αντίστοιχα, ανακατευθύνουν τον χρήστη στις προσωπικές ιστοσελίδες του προγραμματιστή. Το τελευταίο στοιχείο της λίστας αυτής, κάνει ανακατεύθυνση στο e-mail του προγραμματιστή.

8.2 AboutFragment

8.2.1 onCreateView() Method

Ακολουθεί ο κώδικας της μεθόδου όπου γίνεται η σύνδεση των γραφικών του αρχείο fragment_about.xml με την κλάση.

```
public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
    View rootView = inflater.inflate(R.layout.fragment_about, container,
false);
    return rootView;
}
```

Ο κώδικας του .xml αρχείο που περιέχει τα γραφικά.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context=".AboutFragment"
    android:id="@+id/about">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:text="@string/info_title"
        android:id="@+id/info_title"
        android:paddingTop="10dp"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"/>

    <ListView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/info_list"
        android:paddingBottom="20dp"/>

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:text="@string/contact_title"
        android:id="@+id/contact_title"
        android:paddingTop="10dp"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"/>

    <ListView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/contact_list"
        android:paddingBottom="20dp"/>

</LinearLayout>
```

8.2.2 onViewCreated() Method

Σε αυτή την μέθοδο αρχικά δημιουργούνται οι δύο λίστες που βλέπει ο χρήστης όπως φαίνεται παρακάτω.

```
infoListView = (ListView) rootView.findViewById(R.id.info_list);
contactListView = (ListView) rootView.findViewById(R.id.contact_list);

//Creating the "info" listView in the about fragment
String[] info = new String[] {"Help", "About", "Version"};
ArrayList<String> infoList = new ArrayList<String>();
infoList.addAll(Arrays.asList(info));

infoListAdapter = new ArrayAdapter<String>
(getActivity().getApplicationContext(), R.layout.listview_setting_row,
R.id.setting_row, infoList);
infoListView.setAdapter(infoListAdapter);

//Creating the "contact us" listView in the about fragment
String[] contact = new String[] {"Google+", "Facebook", "Email"};
ArrayList<String> contactList = new ArrayList<String>();
contactList.addAll(Arrays.asList(contact));

contactListAdapter = new ArrayAdapter<String>
(getActivity().getApplicationContext(), R.layout.listview_setting_row,
R.id.setting_row, contactList);
contactListView.setAdapter(contactListAdapter);
```

Στη συνέχεια ακολουθεί η αλληλεπίδραση με τον χρήστη. Στα στοιχεία της πρώτης λίστας απλά δημιουργείται ένα dialog παράθυρο που εμφανίζει της διάφορες πληροφορίες.

```
infoListView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position,
long id) {
        String title = "";
        int message = 0;

        switch (position) {
            case 0:
                title = "Help";
                message = R.string.help_info;
                break;
            case 1:
                title = "About";
                message = R.string.about_info;
                break;
            case 2:
                title = "Version";
                message = R.string.version_info;
                break;
            default:
                break;
        }

        //creating of the dialog box
        AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
        builder.setCancelable(true);
        builder.setPositiveButton("OK", null);
        builder.setMessage(message);
        AlertDialog dialog = builder.create();
        dialog.setCanceledOnTouchOutside(true);
        dialog.setTitle(title);
        dialog.show();
    }
});
```

Για την δεύτερη λίστα, όταν ο χρήστης πατήσει πάνω σε κάποιο στοιχείο δημιουργείται ένα Intent και ένα καινούργιο activity για αυτό Intent.

```
contactListView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position,
long id) {
        switch (position) {
            case 0:
                Intent googleIntent = new Intent();
                googleIntent.setAction(Intent.ACTION_VIEW);
                googleIntent.addCategory(Intent.CATEGORY_BROWSABLE);

googleIntent.setData(Uri.parse("https://plus.google.com/u/0/1001930739535835716
29"));
                startActivity(googleIntent);
                break;
```

```
case 1:
                Intent facebookIntent = new Intent();
                facebookIntent.setAction(Intent.ACTION_VIEW);
                facebookIntent.addCategory(Intent.CATEGORY_BROWSABLE);

facebookIntent.setData(Uri.parse("https://www.facebook.com/nikos.mag.3"));
                startActivity(facebookIntent);
                break;
            case 2:
                String[] TO = {"nikosmag7@gmail.com"};
                Intent emailIntent = new Intent(Intent.ACTION_SEND);
                emailIntent.setData(Uri.parse("mailto:"));
                emailIntent.setType("text/plain");
                emailIntent.putExtra(Intent.EXTRA_EMAIL,TO);
                try {
                    startActivity(Intent.createChooser(emailIntent, "Send mail
to creator"));
                    getActivity().finish();
                }
                catch (android.content.ActivityNotFoundException ex) {
                    Toast.makeText(getActivity(), "Email client is dead!",
Toast.LENGTH_SHORT).show();
                }
                break;
            default:
                break;
        }
    }
});
```

Κεφάλαιο 9

9 Κατέβασμα της εφαρμογής

Εάν κάποιος επιθυμεί να εγκαταστήσει την εφαρμογή που περιγράφτηκε στην παρούσα διπλωματική, μπορεί να κατεβάσει το .apk την εφαρμογής από την εξής διεύθυνση

https://www.dropbox.com/s/mhxykasgywlq55g/app-release.apk?dl=0

ή να έρθει σε επικοινωνία με τον προγραμματιστή στέλνοντας μήνυμα στο παρακάτω e-mail.

mailto:maggasia@inf.uth.gr?subject=Personal Digital Agenda

Κεφάλαιο 10

10 Βιβλιογραφία

1. Tutorials σχετικά με την ανάπτυξη εφαρμογών σε Android
   https://www.thenewboston.com

2. Επίσημος οδηγός προγραμματισμού σε Android Studio
   http://developer.android.com/tools/studio/index.html

3. Σχετικά με την βιβλιοθήκη Telerik UI
   http://docs.telerik.com

4. Επίσημος οδηγός προγραμματισμού σε συσκευές Android
   http://developer.android.com/guide/index.html

5. Σχετικά με την ανάπτυξη κώδικα
   http://www.tutorialspoint.com/android/index.htm
   https://stackoverflow.com
   http://www.androidhive.info